Foundations of Applied Combinatorics

Solutions Manual

Edward A. Bender
S. Gill Williamson

Section 1.1

1.1.1. We can form n digit numbers by choosing the leftmost digit AND choosing the next digit AND $\cdots$ AND choosing the rightmost digit. The first choice can be made in 9 ways since a leading zero is not allowed. The remaining $n-1$ choices can each be made in 10 ways. By the Rule of Product we have $9 \times 10^{n-1}$.

To count numbers with at most n digits, we could sum up $9 \times 10^{k-1}$ for $1 \leq k \leq n$. The sum can be evaluated since it is a geometric series. This does not include the number 0. Whether we add 1 to include it depends on our interpretation of the problem's requirement that there be no leading zeroes. There is an easier way. We can pad out a number with less than n digits by adding leading zeroes. The original number can be recovered from any such n digit number by stripping off the leading zeroes. Thus we see by the Rule of Product that there are 10^n numbers with at most n digits. If we wish to rule out 0 (which pads out to a string of n zeroes), we must subtract 1.

1.1.2. The only possible vowel and consonant pattern satisfying the two nonadjacent vowels and initial and terminal consonant conditions is CVCVC. By the Rule of Product, there are $3 \times 2 \times 3 \times 2 \times 3 = 108$ possibilities.

1.1.3. List the elements of the set in any order: $a_1, a_2, \ldots, a_{|S|}$. We can construct a subset by

including a_1 or not AND
including a_2 or not AND
. . .
including $a_{|S|}$ or not.

Since there are 2 choices in each case, the Rule of Product gives $2 \times 2 \times \cdots \times 2 = 2^{|S|}$.

1.1.4 (a) To form a composition of n, we can write n ones in a row and insert either "+" or "," in the spaces between them. This is a series of 2 choices at each of $n-1$ spaces, so we obtain 2^{n-1} compositions of n.

(b) Reversing the roles in a compostion with k parts gives a composition with $n+1-k$ parts. Since this reversal is a one-to-one correspondence between compositions of n and the average number of parts in the two corresponding compostions is $\frac{k+n+1-k}{2}$, we are done.

1.1.5. The answers are SISITS and SISLAL. We'll come back to this type of problem when we study decision trees.

Section 1.2

1.2.1. If we want all assignments of birthdays to people, then repeats are allowed in the list mentioned in the hint. This gives 365^{30}. If we want all birthdays distinct, no repeats are allowed in the list. This gives $365 \times 364 \times \cdots \times (365-29)$. The ratio is 0.29. How can this be computed? There are a lot of possibilities. Here are some.

- Use a symbolic math package.
- Write a computer program.
- Use a calculator. Overflow may be a problem, so you might write the ratio as $(365/365) \times (364/365) \times \cdots \times (336/365)$.
- Use (1.2). You are asked to do this in the next problem. Unfortunately, there is no guarantee how large the error will be.
- Use Stirling's formula after writing the numerator as $365!/335!$. Since Stirling's formula has an error guarantee, we know we are close enough. Computing the values directly from Stirling's

formula may cause overflow. This can be avoided in various ways. One is to rearrange the various factors by using some algebra:

$$\frac{\sqrt{2\pi 365}(365/e)^{365}}{\sqrt{2\pi 335}(335/e)^{335}(365)^{30}} = \sqrt{365/335}\,(365/335)^{335}/e^{30}.$$

Another way is to compute the logarithm of Stirling's formula and use that to estimate the logarithm of the answer.

1.2.2. We want $(n!/(n-k)!)/n^k$ when $n = 365$ and $k = 30$. (See the solution to the previous exercise.) From (1.2),

$$\frac{n!}{(n-k)!\,n^k} \sim e^{-k^2/2n} \quad \text{provided } k = o(n^2/3).$$

Since we need $k = o(n^{2/3})$ and $365^{2/3} \approx 10^{5/3} \approx 50$, our estimate may not be too good. The estimate is $e^{-30^/(2\times 365)} = e^{-900/730} = 0.2915$, which is rather close to the correct answer of 0.2937.

1.2.3. Each of the 7 letters ABMNRST appears once and each of the letters CIO appears twice. Thus we must form an ordered list from the 10 distinct letters. The solutions are

$$\begin{aligned} k = 2:&\quad 10\times 9 = 90\\ k = 3:&\quad 10\times 9\times 8 = 720\\ k = 4:&\quad 10\times 9\times 8\times 7 = 5040 \end{aligned}$$

1.2.4. This can be done in many ways. Some methods lead to lots of cases joined by OR which must be added by the Rule of Sum; other methods lead to a few cases. Here is one of the simplest.

For $k = 2$, the letters are distinct OR equal. By the previous exercise, there are 90 distinct choices. Since the only repeated letters are CIO, there are 3 ways to get equal letters. This gives 93.

For $k = 3$, we have either all distinct OR two equal. The two-equal case can be worked out as follows:

choose the repeated letter (3 ways) AND
choose the positions for the two copies of the letter (3 ways) AND
choose the remaining letter ($10 - 1 = 9$ ways).

By the previous exercise and the Rules of Sum and Product, we have $720 + 3 \times 9 \times 3 = 801$.

For $k = 4$ either all four letters are distinct OR there are just three distinct letters OR there are just two distinct letters. There are 5040 ways to choose all letters distinct. In the second case, we have one repeated letter and two distinct letters. Reasoning as for $k = 3$, we get $3\times 6\times 9\times 8 = 1296$. The last case is a bit trickier. (We'll get into problems associated with distinguishing pairs when we discuss hands of cards later. For now, we'll avoid that problem.) We can proceed as follows:

choose the first letter (3 ways) AND
choose where the second occurence if that letter is (3 ways) AND
choose the other letter (2 ways).

This gives 18 for a grand total of 6354.

1.2.5 (a) Since there are 5 distinct letters, the answer is $5 \times 4 \times 3 = 60$.

(b) Since there are 5 distinct letters, the answer is $5^3 = 125$.

(c) Either the letters are distinct OR one letter appears twice OR one letter appears three times. We have seen that the first can be done in 60 ways. To do the second, choose one of L and T to repeat, choose one of the remaining 4 different letters and choose where that letter is to go, giving $2 \times 4 \times 3 = 24$. To do the third, use T. Thus, the answer is $60 + 24 + 1 = 85$.

1.2.6. For (a) we have $5 \times 4 \times \cdots \times (6-k)$. For (b) we have 5^k. For (c) we omit the details, just noting

For $k=1$, there are 5.	For $k=2$, there are 7.	For $k=4$, there are 286.
For $k=5$, there are 820.	For $k=6$, there are 1920.	For $k=7$, there are 3360.
For $k=8$, there are 3360.	For $k>8$, there are none.	

1.2.7 (a) push, push, pop, pop, push, push, pop, push, pop, pop. Remembering to start with something, say a on the stack: $(a(bc))((de)f)$.

(b) This is almost the same as (a). The sequence is 112211212122 and the last "pop" in (a) is replaced by "push, pop, pop."

(c) $a((b((cd)e))(fg))$; push, push, push, pop, push, pop, pop, push, push, pop, pop, pop; 111010011000.

1.2.8. If we remove the first vote (which must be for the first candidate) and the last vote (which must be for the second candidate), we now have an election where ties are allowed but each candidate receives only $n-1$ votes. Thus the answer is C_{n-1}.

1.2.9. Stripping off the initial R and terminal F, we are left with a list of at most 4 letters, at least one of which is an L. There is just 1 such list of length 1. There are $3^2 - 2^2 = 5$ lists of length 2, namely all those made from E, I and L minus those made from just E and I. Similarly, there are $3^3 - 2^3 = 19$ of length 3 and $3^4 - 2^4 = 65$. This gives us a total of 90.

The letters used are E, F, I, L and R in alphabetical order. To get the word before RELIEF, note that we cannot change just the F and/or the E to produce an earlier word. Thus we must change the I to get the preceding word. The first candidate in alphabetical order is F, giving us RELF. Working backwards in this manner, we come to RELELF, RELEIF, RELEF and, finally, RELEEF.

1.2.10. If there are 4 letters besides R and F, then there is only one R and one F, for a total of 65 spellings by the previous problem. If there are 3 letters besides R and F, we may have R$\cdots$F, R$\cdots$FF or RR$\cdots$F, which gives us $3 \times 19 = 57$ words by the previous problem. We'll say there are 3 RF patterns, namely RF, RFF and RRF. If there 2 letters besides R and F, there are 6 RF patterns, namely the three just listed, RFFF, RRFF and RRRF. This gives us $6 \times 5 = 30$ words. Finally, the last case has the 6 RF patterns just listed as well as RFFFF, RRFFF, RRRFF and RRRRF for a total of 10 patterns. This give us 10 words since the one remaining letter must be L. Adding up all these cases gives us $65+57+30+10 = 162$ possible spellings. Incidentally, there is a simple formula for the number of n long RF patterns, namely $n-1$. Thus there are

$$1+2+\ldots+(n-1) \;=\; n(n-1)/2$$

of length at most n. This gives our previous counts of 1, 3, 6 and 10. The spelling five before RELIEF is REILIF.

1.2.11. There are $n!/(n-k)!$ lists of length k. The total number of lists (not counting the empty list) is

$$\frac{n!}{(n-1)!} + \frac{n!}{(n-2)!} + \cdots + \frac{n!}{1!} + \frac{n!}{0!} \;=\; n!\left(\frac{1}{0!} + \frac{1}{1!} + \cdots + \frac{1}{(n-1)!}\right) \;=\; n!\sum_{i=0}^{n-1} \frac{1^i}{i!}.$$

Since $e \;=\; e^1 \;=\; \sum_{i=0}^{\infty} 1^i/i!$, it follows that the above sum is close to e.

1.2.12 (a) This is just an ordered list, of which there are $n!$.

(b) There are two ways to convert such a seating into one of the type considered in (a): Seat left to right or seat right to left. If the answer is N, this means that $N \times 2 = n!$ and so $N = n!/2$.

(c) Reading along one side and then the other, we get an ordered list and so the answer is $n!$.

(d) Now we can order each side separately and so we have $N \times 2 \times 2 = n!$. Thus $N = n!/4$.

(e) Now, if we switch one side left to right, we must do the same to the other side. This gives us $N \times 2 = n!$ and so $N = n!/2$.

1.2.13. We can only do parts (a) and (d) at present.

(a) A person can run for one of k offices or for nothing, giving $k+1$ choices per person. By the Rule of Product we get $(k+1)^p$.

(d) We can treat each office separately. There are $2^p - 1$ possible slates for an office: any subset of the set of candidates except the empty one. By the Rule of Product we have $(2^p - 1)^k$.

1.2.14. Suppose we have a circular list of prime length n and that the lists obtained by cutting it at two different positions separated by k look the same. Let the list be $a_1, a_2, \ldots, a_n$. Since the two cuts look the same, $a_i = a_{i+k}$ for all i, where subscripts that exceed n are reduced to the range $[1, n]$ by subtracting a multiple of n. Replacing i by $i+k$, we get $a_{i+k} = a_{i+2k}$, and so $a_i = a_{i+2k}$. Repeating this process, we get that $a_i = a_{i+mk}$ for any positive integer m. A result from number theory, which we won't prove, states that when n is a prime and k is not a multiple of n, there is an integer r such that $rk - 1$ is a multiple of n. With $m = r$, we see that $a_i = a_{i+1}$. Thus, if two cuts of the circular permutation give the same linear permutation, all the a's are equal. Thus the number of permutations of length n in which the symbols are not all the same is n times the number of circular permutations of the same sort. With two letters we have $2^n - 2$ such permutations and so there are $(2^n - 2)/n$ such circular permutations for a total of $2 + (2^n - 2)/n$ circular permutations. for L letters, replace 2 by L.

Section 1.3

1.3.1. After recognizing that $k = n\lambda$ and $n - k = n(1 - \lambda)$, it's simply a matter of algebra.

1.3.2. Instead of considering each ballot to be distinct, we can equally well think of two types of votes, namely 1 and 2 depending on who was voted for. Let T_n be the number of ways to order the votes and let A_n (resp. B_n) be the number of ways to order them so that that the first (resp. second) was never behind. The answer is $\frac{A_n + B_n}{T_n}$, where we need to evaluate A_n, B_n T_n. Since the n votes for the first candidate could be in any of the $2n$ positions, $T_n = \binom{2n}{n}$. By Example 1.12, $A_n = B_n = C_n = \frac{1}{n+1}\binom{2n}{n}$, the Catalan numbers. Thus the desired probability is $\frac{2}{n+1}$.

1.3.3. Choose values for pairs AND choose suits for the lowest value pair AND choose suits for the middle value pair AND choose suits for the highest value pair. This gives $\binom{13}{3}\binom{4}{2}^3 = 61,776$.

1.3.4. If we try to keep track of all possible orders (such as card 1, card 2, card paired with 1, card 4, card paired with 4), we will have a lot of cases to consider. There is an easier way. A 5-card hand can be ordered in 5! ways, so this is the number of ways it could be dealt. Thus we simply multiply the number of 2 pair hands by 5!.

1.3.5. Choose the lowest value in the straight (A to 10) AND choose a suit for each of the 5 values in the straight. This gives $10 \times 4^5 = 10240$.

Although the previous answer is acceptable, a poker player may object since a "straight flush" is better than a straight—and we included straight flushes in our count. Since a straight flush is a straight all in the same suit, we only have 4 choices of suits for the cards instead of 4^5. Thus, there are $10 \times 4 = 40$ straight flushes. Hence, the number of straights which are not straight flushes is $10240 - 40 = 10200$.

1.3.6. By Exercise 1.1.4, this is the number of ways to insert $k-1$ commas into $n-1$ positions which is $\binom{n-1}{k-1}$.

1.3.7. This is like Exercise 1.2.3, but we'll do it a bit differently. Note that EXERCISES contains 3 E's, 2 S's and 1 each of C, I, R and X. By the end of Example 1.17, we can use (1.4) with $N = 9$, $m_1 = 3$, $m_2 = 2$ and $m_3 = m_4 = m_5 = m_6 = 1$. This gives $9!/3!\,2! = 30240$.

It can also be done without the use of a multinomial coefficient as follows. Choose 3 of the 9 possible positions to use for the three E's AND choose 2 of the 6 remaining positions to use for the two S's AND put a permutation of the remaining 4 letters in the remaining 4 places. This gives us $\binom{9}{3} \times \binom{6}{2} \times 4!$.

The number of eight letter arrangements is the same. To see this, consider a 9-list with the ninth position labeled "unused."

1.3.8. An arrangement is an ordered list formed from 13 things each used 4 times. Thus we have $N = 52$ and $m_i = 4$ for $1 \le i \le 13$ in (1.4).

1.3.9. Think of the teams as labeled and suppose Teams 1 and 2 each contain 3 men. We can divide the men up in $\binom{11}{3,3,2,2,1}$ ways and the women in $\binom{11}{2,2,3,3,1}$ ways.

We must now count the number of ways to form the ordered situation from the unordered one. Be careful—it's not $4! \times 2$ as it was in the example! Thinking as in the early card example, we start out two types of teams, say M or F depending on which sex predominates in the team. We also have two types of refrees. Thus we have two M teams, two F teams, and one each of an F referee and an M referee. We can order the two M teams (2 ways) and the two F teams (2 ways), so there are only 2×2 ways to order and so the answer is $\binom{11}{3,3,2,2,1}^2 \frac{1}{4}$.

1.3.10 (a) LALALAL, LALALAS, LALALAT, LALALIL.

(b) TSITSAT, TSITSIL, TSITSIS, TSITSIT.

(c) LALSALS, LALSALT, LALSASL, LALSAST.

(d) The possible consonant vowel patterns are CCVCCVC, CCVCVCC, CVCCVCC and CVCVCVC. The first three each contain two pairs of adjacent consonants, one isolated consonant and two vowels. Thus each corresponds to $(3 \times 2)^2 \times 3 \times 2^2$ names. The last has four isolated consonants and three vowels and so corresponds to $3^4 \times 2^3$ names. In total, there are 1944 names.

1.3.11. The theorem is true when $k = 2$ by the binomial theorem with $x = y_1$ and $y = y_2$. Suppose that $k > 2$ and that the theorem is true for $k-1$. Using the hint and the binomial theorem with $x = y_k$ and $y = y_1 + y_2 + \cdots + y_{k-1}$, we have that

$$(y_1 + y_2 + \cdots + y_k)^n = \sum_{j=0}^{n} \binom{n}{j} (y_1 + y_2 + \cdots + y_{k-1})^{n-j} y_k^j.$$

Thus the coefficient of $y_1^{m_1} \cdots y_k^{m_k}$ in this is $\binom{n}{m_k} = n!/(n-m_k)!m_k!$ times the coefficient of $y_1^{m_1} \cdots y_{k-1}^{m_{k-1}}$ in $(y_1 + y_2 + \cdots + y_{k-1})^{n-m_k}$. When $n - m_k = m_1 + m_2 + \cdots + m_{k-1}$ the coefficient is $(n-m_k)!/m_1!m_2! \cdots m_{k-1}!$ and otherwise it is zero by the induction assumption. Multiplying by $\binom{n}{m_k}$, we obtain the theorem for k.

		Values of k					
		0	1	2	3	4	5
V	0	1	0	0	0	0	0
a			↘↓	↘↓	↘↓	↘↓	↘↓
l	1	0	1	0	0	0	0
u			↘↓	↘↓	↘↓	↘↓	↘↓
e	2	0	1	1	0	0	0
s			↘↓	↘↓	↘↓	↘↓	↘↓
	3	0	1	3	1	0	0
o			↘↓	↘↓	↘↓	↘↓	↘↓
f	4	0	1	7	6	1	0
			↘↓	↘↓	↘↓	↘↓	↘↓
n	5	0	1	15	25	10	1

Figure S.1.1 Stirling numbers of the second kind.

Section 1.4

1.4.1. The rows are 1,7,21,35,35,7,1 and 1,8,28,56,70,56,28,8,1.

1.4.2. The recursion makes sense for $k \geq 1$ and $n \geq 1$ if we define

$$S(0,0) = 1 \quad \text{and} \quad S(j,0) = S(0,j) = 0 \quad \text{for} \quad j > 0.$$

Other starting conditions are possible; for example, $S(j,0) = 0$ for $j > 0$, $S(n,n) = 1$ for $n > 0$ and the recursion making sense for $0 < k < n$. (In the latter case, it is understood that the values for $k > n$ are 0.) Figure S.1.1 shows the computation of the values through $n = 5$.

1.4.3. Let $L(n,k)$ be the number of ordered k-lists without repeats that can be made from an n-set S. Form such a list by choosing the first element AND then forming a $k-1$ long list using the remaining $n-1$ elements. This gives $L(n,k) = nL(n-1,k-1)$.

Single out one item $x \in S$. There are $L(n-1,k)$ lists not containing x. If x is in the list, it can be in any of k positions AND the rest of the list can be constructed in $L(n-1,k-1)$ ways. Thus

$$L(n,k) \;=\; L(n-1,k) + kL(n-1,k-1).$$

1.4.4 (a) This can be proved by writing the binomial coefficients in terms of factorials. It can also be proved from the definition of the binomial coefficient: Choosing a set of size k from a set of size n is equivalent to throwing away a set of size $n-k$, namely the things not chosen.

(b) The total number of subsets of an n element set is 2^n. On the other hand, we can divide the subsets into collections T_j, where T_i contains all the i element subsets. The number of subsets in T_i is $\binom{n}{i}$. Apply the Rule of Sum.

(c) The easiest way to prove this is to take the generating function $(1+x)^n = \sum \binom{n}{k} x^k$ and set $x = -1$.

(d) This can be done with generating functions or by a counting argument. For the former approach, write $(1+x)^{n+m} = (1+x)^n(1+x)^m$, expand $(1+x)^n$ and $(1+x)^m$ by the binomial theorem, multiply the results and equate the coefficients of x^k there and in $(1+x)^{n+m}$. For the counting argument, consider disjoint sets N and M with n and m elements respectively. Choose k elements from the union of N and M. The left side counts this directly. The right side breaks this up according to how many of the k elements come from N.

1.4.5. The only way to partition an n element set into n blocks is to put each element in a block by itself, so $S(n,n)=1$. The only way to partition an n element set into one block is to put all the elements in the block, so $S(n,1)=1$.

The only way to partition an n element set into $n-1$ blocks is to choose two elements to be in a block together an put the remaining $n-2$ elements in $n-2$ blocks by themselves. Thus it suffices to choose the 2 elements that appear in a block together and so $S(n,n-1)=\binom{n}{2}$.

The formula for $S(n,n-1)$ can also be proved using (1.8) and induction. The formula is correct for $n=1$ since there is no way to partition a 1-set and have no blocks. Assume true for $n-1$. Use the recurion, the formula for $S(n-1,n-1)$ and the induction assumption for $S(n-1,n-2)$ to obtain

$$S(n,n-1)=S(n-1,n-2)+(n-1)S(n-1,n-1)=\binom{n-1}{2}+(n-1)1=\binom{n}{2},$$

which completes the proof.

Now for $S(n,2)$. Note that $S(n,k)$ is the number of unordered lists of length k where the list entries are nonempty subsets of a given n-set and each element of the set appears in exactly one list entry. We will count ordered lists, which is $k!$ times the number of unordered ones. We choose a subset for the first block (first list entry) and use the remaining set elements for the second block. Since an n-set has 2^n, this would seem to give $2^n/2$; however, we must avoid empty blocks. In the ordered case, there are two ways this could happen since either the first or second list entry could be the empty set. Thus, we must have 2^n-2 instead of 2^n.

Here is another way to compute $S(n,2)$. Look at the block containing n. Once it is determined, the entire two block partition is determined. The block one of the 2^{n-1} subsets of $\underline{n-1}$ with n adjoined. Since something must be left to form the second block, the subset cannot be all of $\underline{n-1}$. Thus there are $2^{n-1}-1$ ways to form the block containing n.

The formula for $S(n,2)$ can also be proved by induction using the recursion for $S(n,k)$ and the fact that $S(n,1)=1$, much as was done for $S(n,n-1)$.

1.4.6. One approach, which we might call “formal,” is to look at the recursion and see what works. We take a second approach, which we might call “combinatorial” or “constructive.” In this approach, we look at the construction and ask what should happen when $k=1$. Since $k=1$, there is only one block, and removing it should leave nothing. Thus, the only term we want is the one with $j=n$ and this should give us 1 since $S(n,1)=1$. To achieve this we want $S(0,0)=1$ and $S(n,0)=0$ when $n\neq 0$.

1.4.7. There are $\binom{n}{k}$ ways to choose the subset AND k ways to choose an element in it to mark. This gives the left side of the recursion times k. On the other hand, there are n ways to choose an element to mark from $\{1,2,\ldots,n\}$ AND $\binom{n-1}{k-1}$ ways to choose the remaining elements of the k-element subset.

1.4.8 (a) We use the hint. Choose i elements of $\{1,2,\cdots,n\}$ to be in the block with $n+1$ AND either do nothing else if $i=n$ OR partition the remaining elements. This gives $\binom{n}{n}$ if $i=n$ and $\binom{n}{i}B_{n-i}$ otherwise. If we set $B_0=1$, the second formula applies for $i=n$, too. Since $i=0$ OR $i=1$ OR $\cdots$ OR $i=n$, the result follows.

(b) We have $B_0=1$ from (a). Using the formula in (a) for $n=0,1,2,3,4$ in order, we obtain $B_1=1$, $B_2=2$, $B_3=5$, $B_4=15$ and $B_5=52$.

1.4.9 (b) Each office is associated with a nonempty subset of the people and each person must be in exactly one subset. This is a partition of the set of candidates with each block corresponding to an office. Thus we have an ordered partition of a n element set into k blocks. The answer is $k!S(n,k)$.

(c) This is like the previous part, except that some people may be missing. We use two methods. First, let i people run for no offices. The remaining $n-i$ can be partitioned in $S(n-i,k)$ ways and the blocks ordered in $k!$ ways. Thus we get $\sum_{i\geq 0}\binom{n}{i}k!S(n-i,k)$. For the second method, either everyone runs for an office, giving $k!S(n,k)$ or some people do not run. In the latter case, we can think of a partition with $k+1$ labeled blocks where the labels are the k offices and "not running." This give $(k+1)!S(n,k+1)$. Thus we have $k!S(n,k)+(k+1)!S(n,k+1)$. The last formula is preferable since it is easier to calculate from tables of Stirling numbers.

(e) Let $T(p,k)$ be the number of solutions. Look at all the people running for the first $k-1$ offices. Let t be the number of these people. If $t<p$, then at least $p-t$ people must be running for the kth office since everyone must run for some office. In addition, any of these t people could run for the kth office. By the Rule of Product, the number of ways we can have this particular set of t people running for the first $k-1$ offices and some people running for the kth office is $T(t,k-1)2^t$. The set of t people can be chosen in $\binom{p}{t}$ ways. Finally, look at the case $t=p$. In this case everyone is running for one of the first $k-1$ offices. The only restriction we must impose is that a nonempty set of candidates must run for the kth office. Putting all this together, we obtain

$$T(p,k) \;=\; \sum_{t=1}^{p-1}\binom{p}{t}T(t,k-1)2^t + T(p,k-1)(2^p-1).$$

This recursion is valid for $p\geq 2$ and $k\geq 2$. The initial conditions are $T(p,1)=1$ for $p>0$ and $T(1,k)=1$ for $k>0$.

Notice that if "people" and "offices" are interchanged, the problem is not changed. Thus $T(p,k)=T(k,p)$ and a recursion could have been obtained by looking at offices that the first $p-1$ people run for. This would give us

$$T(p,k) \;=\; \sum_{t=1}^{k-1}\binom{k}{t}T(p-1,t)2^t + T(p-1,k)(2^k-1).$$

1.4.10. When we speak of a sequence in this exercise, we mean a sequence having no adjacent zeroes whose entries are from $\{0,1,\ldots,d-1\}$. A sequence of length n can be built from a sequence of length $n-1$ by adding something other than zero OR it can be built from a sequence of length $n-1$ that doesn't end in zero by adding zero. (To see this, simply note what is left when you remove the last digit from a sequence of length n.) The sequences of length $n-1$ not ending in zero can all be built by adding something other than zero to a sequence of length $n-2$. Putting this all together by using the Rules of Sum and Product and noting that there are $d-1$ choices for a digit which is not zero, we get $A_n=(d-1)A_{n-1}+(d-1)A_{n-2}$. Since we refer to sequences of length $n-2$, we require that $n\geq 3$. The initial values are $A_1=d$ and $A_2=d^2-1$ since the 2-sequences consist of everything except $0,0$. (If we define $A_0=1$, we could start the recursion at $n=2$.) When $d=10$, the values of A_i for $1\leq i\leq 5$ are 10, 99, 981, 9720 and 96309.

Section 1.5

1.5.1. For each element, there are $j+1$ choices for the number of repetitions, namely anything from 0 to j, inclusive. By the Rule of Product, we obtain $(j+1)^{|S|}$.

1.5.2. When $j=1$, we are simply talking about subsets so the answer is $\binom{|S|}{k}$. Since a k-multiset contains k things, when $j\geq k$, there is no restriction on repetition. Thus we are simply counting unrestricted multisets of which there are $\binom{|S|+k-1}{k}=\binom{|S|+k-1}{|S|-1}$.

1.5.3. To form an unordered list of length k with repeats from $\{1, 2, \ldots, n\}$, either form a list without n OR form a list with n. The first can be done in $M(n-1, k)$ ways. The second can be done by forming a $k-1$ element list AND then adjoining n to it. This can be done in $M(n, k-1) \times 1$ ways. Initial conditions: $M(n, 0) = 1$ for $n \geq 0$ and $M(0, k) = 0$ for $k > 0$.

1.5.4. Let the elements of the set be "place a ball in box i" for $1 \leq i \leq n$. Select k elements and do what they say. Clearly each placement arises exactly once this way.

1.5.5. Interpret the points between the ith and the $(i+1)$st vertical bars as the balls in box i. Since there are $n+1$ bars, there are n boxes. Since there are $(n+k-1) - (n-1) = k$ points, there are k balls.

1.5.6. As in the text, consider a term in the sum obtained by expanding

$$(1 + x_1 + x_1 x_1)(1 + x_2 + x_2 x_2) \cdots (1 + x_n + x_n x_n)$$

using the distributive law. The number of times x_i appears in the term is the number of balls in box i. Thus box i never has more than two balls. The total number of all kinds of x_*'s in the term is the number of balls. Replacing all x_*'s with x's makes the number of x's the number of balls in that particular placement. Suppose each box can contain up to j balls. Instead of $1 + x + x^2$, we now have $1 + x + x^2 + \cdots + x^j$.

1.5.7. This exercise and the previous one are simply two different ways of looking at the same thing since an unordered list with repetitions allowed is the same as a multiset. The nth item must appear zero, one OR two times. The remaining $n-1$ items must be used to form a list of length k, $k-1$ or $k-2$ respectively. This gives the three terms on the left. We generalize to the case where each item is used at most j times: $T(n, k) \ = \ \sum_{i=0}^{j} T(n-1, k-i)$.

1.5.8. We will induct on $n > 0$ and, for each value of n, we will induct on $k > 0$. It is easily verified that $M(1, k) = 1$ for all $k > 0$ and $M(n, 1) = 1$ for all $n > 0$ by the definition of R. This agrees with $\binom{n+k-1}{k}$. Thus we can assume that $n > 1$ and $k > 1$ for the induction step. From Exercise 1.5.3,

$$M(n, k) \ = \ M(n-1, k) + M(n, k-1),$$

and so $M(n, k) \ = \ \binom{(n-1)+k-1}{k} + \binom{n+(k-1)-1}{k-1}$ by the induction hypothesis. The formula $\binom{a}{k} + \binom{a}{k-1} \ = \ \binom{a+1}{k}$ completes the proof.

1.5.9 (a) We give two solutions. Both use the idea of inserting a ball into a tube in an arbitrary position. To physically do this may require some manipulation of balls already in the tube.

1. Insert $b-1$ balls into the tubes AND then insert the b^{th} ball. There are $i+1$ possible places to insert this ball in a tube containing i balls. Summing this over all t tubes gives us $(b-1)+t$ possible places to insert the b^{th} ball. We have proved that

$$f(b, t) \ = \ f(b-1, t)(b+t-1).$$

Since $f(1, t) = t$, we can establish the formula by induction.

2. Alternatively, we can insert the first ball AND insert the remaining $b-1$ balls. The first ball has the effect of dividing the tube in which it is placed into two tubes: the part above it and the part below. Thus

$$f(b, t) \ = \ t f(b-1, t+1),$$

and we can again use induction.

(b) We give two solutions:

Construct a list of length $t+b-1$ containing each ball exactly once and containing $t-1$ copies of "between tubes." This can be done in $\binom{t+b-1}{t-1}b!$ ways—choose the "between tubes" and then permute the balls to place them in the remaining b positions in the list.

Alternatively, imagine an ordered $b+t-1$ long list. Choose $t-1$ positions to be divisions between tubes AND choose how to place the b balls in the remaining b positions. This gives $\binom{b+t-1}{t-1} \times b!$.

1.5.10 (a) $f(n,k) = f(n-1,k-1) + (n+k-1)f(n-1,k)$.

(b) If the order of the blocks also mattered, the number of solutions would be $f(n,k)\,k!$. On the other hand, we can obtain such partitions by constructing an ordered list of the n things and then choosing $k-1$ places between them (without replacement) to be the divisions between boxes. This gives us a count of $n!\,\binom{n-1}{k-1}$.

Section 2.1

2.1.2 (a) Since f is an injection, every element of A maps to a different element of B. Thus B must have at least as many elements as A.

(b) Since f is a surjection, every element of B is the image of at least one element of A. Thus A must have at least as many elements as B.

(c) Combine the two previous results.

(d) Suppose that f is an injection and not a surjection. Then there is some $b \in B$ which is not the image of any element of A under f. Hence f is an injection from A to $B - \{b\}$. By (a), $|A| \leq |B - \{b\}| < |B|$, contradicting $|A| = |B|$.
Now suppose that f is a surjection and not an injection. Then there are $a, a' \in A$ such that $f(a) = f(a')$. Consider the function f with domain restricted to $A - \{a'\}$. It is still a surjection to B and so by (b) $|B| \leq |A - \{a'\}| < |A|$, contradicting $|A| = |B|$.

(e) By the previous part, if f is either an injection or a surjection, then it is both, which is the definition of a bijection.

Section 2.2

2.2.2. Imagine writing the permutation in cycle form. Look at the cycle containing 1, starting with 1. There are $n-1$ choices for the second element of the cycle AND then $n-2$ choices for the third element AND $\cdots$ AND $(n-k+1)$ choices for the kth element.

(a) The answer is given by the Rule of Product and the above result with $k = n$.

(b) We write the cycle containing 1 in cycle form as above AND then permute the remaining $n-k$ elements of $\underline{n}$ in any fashion. For the k long cycle containing 1, the above result gives $\frac{(n-1)!}{(n-k)!}$ choices. There are $(n-k)!$ permutations on a set of size $n-k$. Putting this all together using the Rule of Product, we get $(n-1)!$, a result which does not depend on k.

(c) Since each permutation has probability $1/n!$, this follows immediately from the previous parts.

2.2.3. The interchanges can be written as (1,3), (1,4) and (2,3). Thus the entire set gives $1 \to 3 \to 2$, $2 \to 3$, $3 \to 1 \to 4$ and $4 \to 1$. In cycle form this is (1,2,3,4). Thus five applications takes 1 to 2.

2.2.4. We look at $P_k(n+1)$. Suppose $n+1$ is in a cycle with j other elements. Then we must have $0 \le j < k$. The j elements can be chosen in $\binom{n}{j}$ ways, AND the $j+1$ elements can be formed into a $(j+1)$-cycle in $j!$ ways, AND the remaining $n-j$ elements of $\underline{n+1}$ can be arranged into cycles in $P_k(n-j)$ ways. By the Rules of Sum and Product

$$P_k(n+1) \;=\; \sum_{j=0}^{k-1} \binom{n}{j} j! P_k(n-j).$$

We claim this is valid for $n \ge 0$. To see this, note two things: First, if the term $j = n$ occurs, there are no $n-j$ elements to arrange and so we should not have the factor $P_k(0)$. Since $P_k(0) = 1$, that accomplishes the same thing as dropping the factor $P_k(0)$. Second any terms with $j > n$ should be removed since this is impossible. When $j > n$, $\binom{n}{j} = 0$ and so those terms are all zero.

2.2.5 (a) This was done in Exercise 2.2.2, but we'll redo it. If $f(k) = k$, then the elements of $\underline{n} - \{k\}$ can be permuted in any fashion. This can be done in $(n-1)!$. Since there are $n!$ permutations, the probability that $f(k) = k$ is $(n-1)!/n! = 1/n$. Hence the probability that $f(k) \ne k$ is $1 - 1/n$.

(b) By the independence assumption, the probability that there are no fixed points is $(1 - 1/n)^n$. One of the standard results in calculus is that this approaches $1/e$ as $n \to \infty$. (You can prove it by writing $(1-1/n)^n = \exp(\ln(1-1/n)/(1/n))$, setting $1/n = x$ and using l'Hôpital's Rule.)

(c) Choose the k fixed points AND construct a derangement of the remaining $n-k$. This gives us $\binom{n}{k} D_{n-k}$. Now use $D_{n-k} \approx (n-k)!/e$.

2.2.6 (a) To construct a permutation with k cycles, break it into cases according to the number i of elements in the cycle with 1. For each such case, choose those i elements AND then construct a cycle containing them and 1 AND construct a permutation of the remaining $n-i$ elements that contains exactly $k-1$ cycles.

(b) It is valid for all positive n and k if we set $z(0,0) = 1$ and set $z(0,j) = z(j,0) = 0$ for $j \ne 0$.

(c) The rows correspond to values of n and the columns to values of k.

	1	2	3	4	5
1	1	0	0	0	0
2	1	1	0	0	0
3	2	3	1	0	0
4	6	11	6	1	0
5	24	50	35	10	1

2.2.7. For $1 \le k \le n-1$, $\mathbf{E}(|a_k - a_{k+1}|) = \mathbf{E}(|i-j|)$, where the latter expectation is taken over all $i \ne j$ in $\underline{n}$. Thus the answer is $(n-1)$ times the average of the $n(n-1)$ values of $|i-j|$ and so

$$\begin{aligned}
\text{answer} &= \frac{n-1}{n(n-1)} \sum_{i \ne j} |j-i| \;=\; \frac{n-1}{n(n-1)} \sum_{i,j} |j-i| \;=\; \frac{2}{n} \sum_{1 \le i \le j \le n} (j-i), \qquad \text{proving (a)} \\
&= \frac{2}{n} \sum_{j=1}^{n} \sum_{i=1}^{j} (j-i) \;=\; \frac{2}{n} \sum_{j=1}^{n} \left(j^2 - \frac{j(j+1)}{2} \right) \;=\; \frac{1}{n} \sum_{j=1}^{n} (j^2 - j) \\
&= \frac{1}{n} \left(\frac{n(n+1)(2n+1)}{6} - \frac{n(n+1)}{2} \right) \;=\; \frac{n^2-1}{2}.
\end{aligned}$$

Section 2.3

2.3.2 (a) The coimage of a function is a partition of the domain with one block for each element of Image(f).

(b) You can argue this directly or apply the previous result. In the latter case, note that since Coimage(f) is a partition of A, $|\,\text{Coimage}(f)| = |A|$ if and only if each block of Coimage(f) contains just one element. On the other hand, f is an injection if and only if no two elements of A belong to the same block of Coimage(f).

(c) By the first part, this says that $|\,\text{Image}(f)| = |B|$. Since Image($f$) is a subset of B, it must equal B.

2.3.3. We can form the permutations of the desired type by first constructing a partition of $\underline{n}$ counted by $B(n,\vec{b})$ AND then forming a cycle from each block of the partition. The argument used in Exercise 2.2.2 proves that there are $(k-1)!$ cycles of length k that can be made from a k-set.

2.3.4. The sequence must be $a_1 < \cdots < a_k > \cdots > a_{2k-1}$ for some $k > 0$. Let $a_k = t$. Then there are $\binom{t-1}{k-1}$ possibilities for $a_1, \ldots, a_{k-1}$. The same answer holds for $a_{k+1}, \ldots, a_{2k-1}$. Thus we have

$$\sum_{t=1}^{n}\sum_{k=1}^{\infty}\binom{t-1}{k-1}^2 = \sum_{t=1}^{n}\sum_{k=1}^{t}\binom{t-1}{k-1}^2$$

Using results from Exercise 1.4.4 (p. 32), we can simplify this:

$$\sum_{k=1}^{t}\binom{t-1}{k-1}^2 = \sum_{i=0}^{t-1}\binom{t-1}{i}^2 = \sum_{i=0}^{t-1}\binom{t-1}{i}\binom{t-1}{t-1-i} = \binom{2t-2}{t-1},$$

and so the answer is

$$\sum_{t=1}^{n}\binom{2t-2}{t-1} = \sum_{i=0}^{n-1}\binom{2i}{i}.$$

2.3.5 (a) In the order given, they are 2, 1, 3 and 4

(b) If f is associated with a B partition of $\underline{n}$, then B is the coimage of f and so f determines B.

(c) See (b).

(d) The first is not since $f(1) = 2 \neq 1$.
The second is: just check the conditions.
The third is not since $f(4) - 1 = 2 > \max(f(1), f(2), f(3)) = 1$.
The fourth is: just check the conditions.

(e) In a way, this is obvious, but it is tedious to write out a proof. By definition $f(1) = 1$. Choose $k > 1$ such that $f(x) = k$ for some x. Let y be the least element of $\underline{n}$ for which $f(y) = k$. By the way f is constructed, y is not in the same block with any $t < y$. Thus y is the smallest element in its block and so $f(y)$ will be the smallest number exceeding all the values that have been assigned for $f(t)$ with $t < y$. Thus the maximum of $f(t)$ over $t < y$ is $k - 1$ and so f is a restricted growth function.

(f) The functions are given in one-line form and the partition below them

1 1 1 1 $\{1,2,3,4\}$	1 1 1 2 $\{1,2,3\}\,\{5\}$	1 1 2 1 $\{1,2,4\}\,\{3\}$	1 1 2 2 $\{1,2\}\,\{3,4\}$	1 1 2 3 $\{1,2\}\,\{3\}\,\{4\}$
1 2 1 1 $\{1,3,4\}\,\{2\}$	1 2 1 2 $\{1,3\}\,\{2,4\}$	1 2 1 3 $\{1,3\}\,\{2\}\,\{4\}$	1 2 2 1 $\{1,4\}\,\{2,3\}$	1 2 2 2 $\{1\}\,\{2,3,4\}$
1 2 2 3 $\{1\}\,\{2,3\}\,\{4\}$	1 2 3 1 $\{1,4\}\,\{2\}\,\{3\}$	1 2 3 2 $\{1\}\,\{2,4\}\,\{3\}$	1 2 3 3 $\{1\}\,\{2\}\,\{3,4\}$	1 2 3 4 $\{1\}\,\{2\}\,\{3\}\,\{4\}$

2.3.6. If no block contained more than $(|S|-1)/k$ elements, the number of elements would be at most

$$\frac{|S|-1}{k}\,k \;=\; |S|-1 \;<\; |S|,$$

which cannot be.

2.3.7. The coimage is a partition of A into at most $|B|$ blocks, so our bound is $1+(|A|-1)/|B|$.

2.3.8. There are only n possible values of m, so some value must occur more than $((n+1)-1)/n = 1$ times. Given two numbers with the same m, one divides the other.

2.3.9. If $s < t$ and $f(s) = f(t)$, that tells us that we cannot put a_s at the start of the longest decreasing subsequence starting with a_t to obtain a decreasing subsequence. (If we could, we'd have $f(s) \geq f(t)+1$.) Thus, $a_s > a_t$. Hence the subsequence $a_i, a_j, \ldots$ constructed in the problem is increasing.

Now we're ready to start the proof. If there is a decreasing subsequence of length $n+1$ we are done. If there is no such subsequence, $f : \underline{\ell} \to \underline{n}$. By the generalized Pigeonhole Principle, there is sum k such that $f(t) = k$ for at least ℓ/n values of t. Thus it suffices to have $\ell/n > m$. In other words $\ell > mn$.

2.3.10. Suppose S contains p pairs of integers. If $p > N$, then two sums must have the same remainder when divided by N since only N remainders are possible.

If the numbers in a pair must be distinct, there are $\binom{t}{2}$ pairs and so we need $N < \binom{t}{2} = \frac{t(t-1)}{2}$. Solving this quadratic inequality, we obtain $t > \frac{1+\sqrt{1+8N}}{2}$ or $t < \frac{1-\sqrt{1+8N}}{2}$. The latter solution makes no sense since we know $t > 0$.

If the pair may contain two copies of the same number, there are $\binom{t}{2} + t = \frac{t(t+1)}{2}$ pairs and we obtain $t > \frac{-1+\sqrt{1+8N}}{2}$.

2.3.11. Let the elements be $s_1, \ldots, s_n$, let $t_0 = 0$ and let $t_i = s_1 + \ldots + s_i$ for $1 \leq i \leq n$. By the Pigeonhole Principle, two of the t's have the same remainder on division by n, say t_j and t_k with $j < k$. It follows that $t_k - t_j = s_{j+1} + \ldots + s_k$ is a multiple of n.

Section 2.4

2.4.1. $x(x+y) = xx + xy = x + xy = x$.

2.4.2. Here is a truth table proof.

x	y	x'	y'	$x \oplus y$	$x' \oplus y'$
0	0	1	1	0	0
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

2.4.3. We state the laws and whether they are true or false. If false we give a counterexample.

(a) $x + (yz) \;=\; (x+y)(x+z)$ is true. (Proved in text.)

(b) $x(y \oplus z) \;=\; (xy) \oplus (xz)$ is true.

(c) $x + (y \oplus z) \;=\; (x+y) \oplus (x+z)$ is false with $x = y = z = 1$.

(d) $x \oplus (yz) \;=\; (x \oplus y)(x \oplus z)$ is false with $x = y = 1$, $z = 0$.

2.4.4. It suffices to show that a NAND can be written using NORs. We have $u' = \mathrm{NOR}(u)$ and

$$\mathrm{NAND}(x, y, \ldots, z) \;=\; \Big(\mathrm{NOR}(x', y', \ldots, z')\Big)'.$$

2.4.5. We use algebraic manipulation. Each step involves a simple formula, which we will not bother to mention. You could also write down the truth table, read off a disjunctive normal form and try to reduce the number of terms.

(a) $(x \oplus y)(x + y) \;=\; (xy' + x'y)(x + y) \;=\; xy' + x'yx + xy'y + x'y \;=\; xy' + x'y$. Note that this is $x \oplus y$.

(b) $(x + y) \oplus z \;=\; (x + y)z' + (x + y)'z \;=\; xz' + yz' + x'y'z$.

(c) $(x + y + z) \oplus z \;=\; (x + y + z)z' + (x + y + z)'z \;=\; xz' + yz' + x'y'z'z \;=\; xz' + yz'$.

(d) $(xy) \oplus z \;=\; xyz' + (xy)'z \;=\; xyz' + x'z + y'z$.

2.4.6. This is the function c in Figure 2.1.

2.4.7. There are many possible answers. A complicated one comes directly from the truth table and contains 8 terms. The simplest form is $xw + yw + zw + xyz$. This can be obtained as follows. $(x+y+z)w$ will give the correct answer except when $x = y = z = 1$ and $w = 0$. Thus we could simply add the term $xyzw'$. By noting that it is okay to add xyz when $w = 1$, we obtain $(x+y+z)w+xyz$.

2.4.8. There are many possible answers. If we note that zw gives the correct answer unless $zw = 0$, $x \neq y$ and at least one of z and w is one, we obtain

$$zw + (xy' + x'y)(z + w) \;=\; zw + xy'z + xy'w + x'yz + x'yw.$$

Section 3.1

3.1.1. From the figures in the text, we see that they are 123, 132 and 321.

3.1.2. We will not draw the tree.

(a) 8 and 19.

(b) 1432 and 3241.

3.1.3. We will not draw the tree. The root is 1, the vertices on the next level are 21 and 12 (left to right). On the next level, 321, 231, 213, 312, 132, and 123. Finally, the leaves are 4321, 3421, 3241, 3214, 4231, 2431, 2341, 2314, 4213, 2413, 2143, 2134, and so on.

(a) 7 and 16.

(b) 2,4,3,1 and 3,1,2,4.

3.1.4. We will not draw the tree. The root is 1, the vertices on the next level are 21 and 12 (left to right). On the next level, 312, 231, 213, 321, 132, and 123 Finally, the leaves are 4123, 3421, 3142, 3124, 4312, 2413, 2341, 2314, 4132, 2431, 2143, 2134 4213, 3412, 3241, 3214, 4321, 1423, 1342, 1324, 4231, 1432, 1243, 1234.

(a) 7 and 8.

(b) 2413 and 3214.

3.1.5. We will not draw the tree. There are nine sequences: ABABAB, ABABBA, ABBABA, ABBABB, BABABA, BABABB, BABBAB, BBABAB and BBABBA.

3.1.6. We will not draw the tree.

(a) 3 and 10.

(b) 4,3,2,1 and 6,4,3,1.

(c) 6,5,4,3 has rank 14.

(e) The decision tree corresponds to 4 element subsets of $\underline{6}$. The leaf 5431 corresponds to the subset $\{5, 4, 3, 1\}$.

3.1.7. We will not draw the tree.

(a) 5 and 18.

(b) 111 and 433.

(c) 4,4,4 has rank 19.

(e) The decision tree for the strictly decreasing functions is interspersed. To find it, discard the leftmost branch leading out of each vertex except the root and then discard those decisions that no longer lead to a leaf of the original tree.

3.1.8. We don't want to draw the tree since we saw that it had 648 leaves in Section 1.1. It is somewhat irregular. Usually it has 4 choices after a consonant and 3 after a vowel. This is not always true though. In summary, the tree does not appear to have a very nice structure.

3.1.9. We assume that you are looking at decision trees in the following discussion.

(a) The permutation of rank 0 is the leftmost one in the tree and so each element is inserted as far to the left as possible. Thus the answer is $n, (n-1), \ldots, 2, 1$.

The permutation of rank $n! - 1$ is the rightmost one in the tree and so each element is inserted as far to the right as possible. Thus the answer is $1, 2, 3, \ldots, n$.

We now look at $n!/2$. Note that the decision about where to insert 2 splits the tree into two equal pieces. We are interested in the leftmost leaf of the righthand piece. The righthand piece means we take the branch $1, 2$. To stay to the left after that, 3 through n are inserted in the leftmost position. Thus the permutation is $n, (n-1), \ldots, 4, 3, 1, 2$.

(b) The permutation of rank 0 is the leftmost one in the tree and so each element is inserted as far to the left as possible. It begins 2,1. Then 3 "bumps" 2 to the end: 3,1,2. Next 4 "bumps" 3 to the end: 4,1,2,3. In general, we have $n, 1, 2, 3, \ldots, (n-1)$.

The permutation of rank $n! - 1$ is the rightmost one in the tree and so each element is inserted as far to the right as possible. Thus the answer is $1, 2, 3, \ldots, n$.

We now look at $n!/2$. Note that the decision about where to insert 2 splits the tree into two equal pieces. We are interested in the leftmost leaf of the righthand piece. The righthand piece means we take the branch $1, 2$. To stay to the left after that, 3 through n are inserted in the leftmost position. This leads to "bumping" as it did for rank 0. Thus the permutation is $n, 2, 1, 3, 4, 5, \ldots, (n-1)$.

(c) You should be able to see that the permutation $(1, 2, 3, \ldots, n)$ has rank 0 in both cases and that the permutation $(n, \ldots, 3, 2, 1)$ has rank $n! - 1$ in both cases.

First suppose that $n = 2m$, an even number. It is easy to see how to split the tree in half based on the first decision as we did for insertion order: Choose $m + 1$ and then stay as left as possible. This means everything is in order except for $m + 1$. Thus the permutation is $m + 1$ followed by the elements of $\underline{n} - \{m + 1\}$ in ascending order.

Now suppose that $n = 2m - 1$. In this case, we must make the middle choice, m and split the remaining tree in half, going to the leftmost leaf of the right part. If you look at some trees, you should see that this leads to the permutation $m, m + 1$ followed by the elements of $\underline{n} - \{m, m + 1\}$ in ascending order.

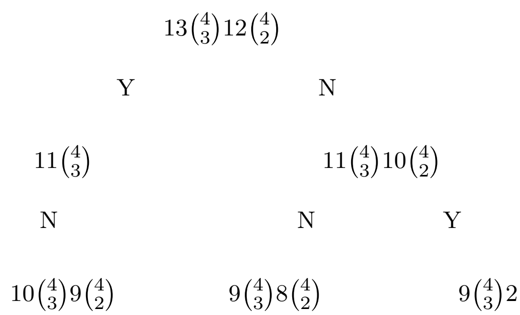


Figure S.3.1 The decision tree for forming three full houses. The numbers at each vertex in the k^{th} level are the number of ways to form the k^{th} hand. Going from the root to a leaf involves two ANDs. The total number of possibilities is about 1.8×10^{10}.

3.1.10. By Example 1.14 (p. 18), the first full house can be formed in 3,744 ways. For the second full house, we must decide whether or not its pair has the same value as the pair in the first full house. (This is such a simple situation that we don't really need a decision tree.) The number of choices for the second hand is

$$11\binom{4}{3} \times 10\binom{4}{2} + 11\binom{4}{3} \;=\; 2{,}640 + 44 \;=\; 2{,}684$$

and the two hands together can be formed in 10,048,896 ways. If the order of the hands does not matter, this should be divided by 2.

3.1.11 (a) We'll make a decision based on whether or not the pair in the full house has the same face value as a pair in the second hand. If it does not, there are

$$\binom{11}{2}\binom{4}{2}^2(52-8-5) \;=\; 77{,}220$$

possible second hands. If it does, there are

$$11\binom{4}{2}(52-8-3) \;=\; 2{,}706$$

possible second hands. Adding these up and multiplying by the number of possible full houses (79,926) gives us about 3×10^8 hands.

(b) There are various ways to do this. The decision trees are all more complicated than in the previous part.

(c) The order in which things are done can be very important.

3.1.12. As usual, we'll form the hands sequentially. The first decision will be whether or not the first and second hands have pairs with the same face value. The second decision will be whether or not the pair in the third hand has a face value that is the same as an earlier pair. We obtain the tree in Figure S.3.1.

3.1.13. You can simply modify the decision tree in Figure 3.5 as follows: Decrease the "number of singles" values by 1 (since the desired word is one letter shorter). Throw away those that become negative; i.e., erase leaves C and H. Add a new path that has no triples, one pair and five singles. Call the new leaf X. It is then neccessary to recompute the numbers. Here are the results, which total to 113,540:

$$\begin{aligned}
X: &\quad \binom{2}{0}\binom{5}{1}\binom{5}{5}\binom{8}{2,\,1,\,1,\,1,\,1,\,1} = 12{,}600\\
A: &\quad \binom{2}{0}\binom{5}{2}\binom{4}{3}\binom{8}{2,\,2,\,1,\,1,\,1} = 50{,}400\\
B: &\quad \binom{2}{0}\binom{5}{3}\binom{3}{1}\binom{8}{2,\,2,\,2,\,1} = 18{,}900\\
D: &\quad \binom{2}{1}\binom{4}{0}\binom{5}{4}\binom{8}{3,\,1,\,1,\,1,\,1} = 8{,}400\\
E: &\quad \binom{2}{1}\binom{4}{1}\binom{4}{2}\binom{8}{3,\,2,\,1,\,1} = 20{,}160\\
F: &\quad \binom{2}{1}\binom{4}{2}\binom{3}{0}\binom{8}{3,\,2,\,2} = 2{,}520\\
G: &\quad \binom{2}{2}\binom{3}{0}\binom{4}{1}\binom{8}{3,\,3,\,1} = 560.
\end{aligned}$$

Section 3.2

3.2.1. We use the rank formula in the text and, for unranking, a greedy algorithm.

(a) $\binom{10}{3}+\binom{5}{2}+\binom{3}{1} = 133.\qquad \binom{8}{4}+\binom{5}{3}+\binom{2}{2}+\binom{0}{1} = 81.$

(b) We have $35=\binom{7}{4}$ so the first answer is 8,3,2,1. The second answer is 12,9,6,5 because

$$\begin{aligned}
\tbinom{11}{4} \le 400 &< \tbinom{12}{4} & 400-\tbinom{11}{4} &= 70\\
\tbinom{8}{3} \le 70 &< \tbinom{9}{3} & 70-\tbinom{8}{3} &= 14\\
\tbinom{5}{2} \le 14 &< \tbinom{6}{2} & 14-\tbinom{5}{2} &= 4\\
\tbinom{4}{1} \le 4 &< \tbinom{5}{1}.
\end{aligned}$$

(c) 9,6,4,2,1 and 9,7,2,1.

(d) 9,5,4,3,2 and 9,6,5,3.

3.2.2. We use the rank formula in the text and, for unranking, a greedy algorithm.

(a) 635124: The decisions are 0,2,0,3,5 and so the rank is

$$0\times 6!/2! + 2\times 6!/3! + 0\times 6!/4! + 3\times 6!/5! + 5\times 6!/6! = 263.$$

4,5,6,1,2,3: The decisions are 0,0,3,3,3 and the rank is 111.

(b) For rank 151 we have 151/(6!/2!) is 0 remainder 151, 151/(6!/3!) is 1 remainder 31, 31/(6!/4!) is 1 remainder 1, 1/(6!/5!) is 0 remainder 1 and 1/(61/6!) is 1 remainder 0. Thus the decision

sequence is 0,1,1,0,1 and the permutation is 1,3,4,2,6,5.
For rank 300, this procedure gives decision sequence 0,2,2,0,0 and permutation 3,4,1,2,5,6.

(c) The answers are 1,2,3,4,6,5,7,8,9 and 6,5,4,1,2,3,7,9,8.

(d) The answers are 8,9,7,1,2,3,4,5,6 and 9,8,7,5,6,4,1,2,3.

3.2.3. One can compute the ranks by looking at the decision tree or by using the formula in Theorem 3.3. We choose the latter approach. In case (j), we have $f(i) = k + j - i$. (This is easily checked since this f clearly decreases by 1 as i increases by 1 and it gives $f(1) = k$, $k+1$ and $k+2$ for $j = 1$, 2 and 3, respectively.) By the theorem,

$$\text{RANK}(f) \;=\; \sum_{i=1}^{k} \binom{f(i)-1}{k+1-i} \;=\; \sum_{i=1}^{k} \binom{k+j-i-1}{k+1-i}$$

When $j = 1$, all the binomial coefficients are 0 and so the answer for the first function is 0.
When $j = 2$, all the binomial coefficients are 1 and so the answer for the second function is k.
When $j = 3$, we have

$$\text{RANK}(f) \;=\; \sum_{i=1}^{k} \binom{k+2-i}{k+1-i} \;=\; \sum_{i=1}^{k} (k+2-i) \;=\; (k+1) + (k) + (k-1) + \cdots + (2).$$

Since the sum of the first n positive integers is $\frac{n(n+1)}{2}$, the rank is $\frac{(k+1)(k+2)}{2} - 1 = \frac{k(k+3)}{2}$.

3.2.4 (a) We give two proofs.
First proof. Since the nonincreasing functions correspond to choosing unordered samples without repetition, the number of them in $\underline{n}^{\underline{k}}$ is $\binom{n+k-1}{k}$. Other than this change from $\binom{n}{k}$, the method for proving Theorem 3.3 works and so we have $\text{RANK}(f) \;=\; \sum_{i=1}^{k} \binom{f(i)+k-i-1}{k-i+1}$.
Second proof. By Example 2.11 (p. 48), there is a bijection between strictly decreasing and nonincreasing functions. The bijection φ in that example preserves the lex order of functions and hence their rank. Apply the bijection to f and then use Theorem 3.3.

(b) 5,5,4,2,1,1: $\binom{9}{6} + \binom{8}{5} + \binom{6}{4} + \binom{3}{3} + \binom{1}{2} + \binom{0}{1} \;=\; 156$.
6,3,3: $\binom{7}{3} + \binom{3}{2} + \binom{2}{1} \;=\; 40$.

(c) As with strictly decreasing functions, we find that $35 \;=\; \binom{7}{4} + \binom{2}{3} + \binom{1}{2} + \binom{0}{1}$ and that $400 \;=\; \binom{11}{4} + \binom{8}{3} + \binom{5}{2} + \binom{4}{1}$. Thus the functions are 5,1,1,1 and 9,7,5,5.

3.2.5 (a) $D_1 \times (n-1)! + D_2 \times (n-2)! + \cdots + D_{n-1} \times 1! \;=\; \sum_{k=1}^{n-1} D_k (n-k)!$.

(b) Denote the permutation by f. Let $L = \underline{n}$. For $i = 1, 2, \ldots, n-1$ in order: let D_i is the number of elements in L which are less than $f(i)$ and replace L with $L - \{f(i)\}$.

(c) The decision sequences are 4,4,0,1,1 and 5,1,2,0,0 and so the ranks are 579 and 636.

(d) By a greedy algorithm we get the decision sequences 1,1,1,0,1 and 2,2,2,0,0. The permutations are 2,3,4,1,6,5 and 3,4,5,1,2,6.

3.2.8. 4; 14; 25; 67; 102.

3.2.9. $00000000000000000000 = 0^{20}$; $11000000000000000000 = 1^2 0^{18}$; 0100; 10101100.

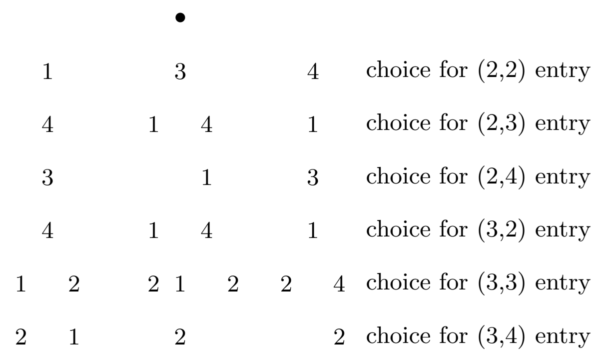


Figure S.3.2 The decision tree for 4×4 standard Latin Squares in Exercise 3.3.1.

3.2.10. If we can see some sort of pattern, we might figure out what to do. Let's make a list of the interchanges. A k means that positions k and $k+1$ are interchanged. Thus going from the first to the second permutation in the list has the interchange 3. The list of interchanges is 3, 2, 1, 3, 1, 2, 3, 1 repeated three times, where the last interchange goes from the last element (2,1,3,4) to the first (1,2,3,4).

That doesn't seem to help. Can we see anything else? Notice how the 4 moves step-by-step from right to left, pauses, moves step-by-step from left to right, pauses (this is the interchange from the bottom of one column to the top of the next), and then repeats the pattern twice. Let's list the interchanges when 4 pauses, dropping 4 from the list:

$$1,2,3 \quad 1,3,2 \quad 3,1,2 \quad 3,2,1 \quad 2,3,1 \quad 2,1,3.$$

It turns out this is the key: 1, 2 and 3 go through all their possible permutations and 4 simply moves through! We can do this with $n = 5$:

1. Start with 1,2,3,4,5
2. Move 5 step-by-step to the left so it is in the first position, say $5, a_1, a_2, a_3, a_4$.
3. The we look in the list for $n = 4$ to find what follows a_1, a_2, a_3, a_4, say b_1, b_2, b_3, b_4.
4. After $5, a_1, a_2, a_3, a_4$, put $5, b_1, b_2, b_3, b_4$.
5. Move 5 step-by-step to the rightt so it is in the last position, say $c_1, c_2, c_3, c_4, 5$.
6. The we look in the list for $n = 4$ to find what follows c_1, c_2, c_3, c_4, say d_1, d_2, d_3, d_4.
7. After $c_1, c_2, c_3, c_4, 5$, put $d_1, d_2, d_3, d_4, 5$.
8. Go to Step 2.

You should see how to generalize this.

Section 3.3

3.3.1. When building and $n \times n$ Latin Square, if the first $n-1$ rows have been filled in, then the last row is determined. Thus we'll omit it from the decision tree. The tree is shown in Figure S.3.2.

3.3.2. The work in drawing the decision tree can be reduced by assuming that, possibly by rotating and flipping the board, the queen in the first row is as far to the right as possible. We will simply list the solutions in the form $c_1, \ldots, c_n$ which means there is a queen in square (i, c_i) for $1 \le i \le n$.

- $n = 4$ has the two solutions 2,4,1,3 and 3,1,4,2.

- $n = 5$ has 10 solutions. Eight come from 3,5,2,4,1 and its rotations and reflections. The other two are 4,1,3,5,2 and its reflection.
- $n = 6$ has four solutions: 3,6,2,5,1,4; 4,1,5,2,6,3; 2,4,6,1,3,5 and 5,3,1,6,4,2.
- $n = 7$ has 40 solutions.
- $n = 8$ has 92 solutions.

3.3.3. You should find 14 solutions.

3.3.4. You should find 8 solutions. They are all obtained by rotating and flipping the solution that has the L-shape in positions (1,1), (1,2) and (2,2). The positions of the 3 dominoes are then forced.

Section 4.1

4.1.1. The Venn diagrams each consist of two intersecting circles.

(a) $V_2 \cap V_3$ contains words of the form $CVVC$. We are interested in $V_2 \cup V_3$, the union of the circles. Thus

$$\begin{aligned} |V_2 \cup V_3| &= |V_2| + |V_3| - |V_2 \cap V_3| \\ &= 21^2 \times 5 \times 26 + 21^2 \times 5 \times 26 - 21^2 \times 5^2 \\ &= 21^2 \times 5 \times 47 \end{aligned}$$

(b) We want all 4 letter words beginning and ending with consonants that are not in $C_2 \cap C_3$, which is $21^2 \times 26^2 - 21^4$.

4.1.2. These are simply derangements: Label each married pair. Assign each person the same label as given to the pair. Let $f(k)$ be the label of the man who is paired with the woman whose label is k. This function is a derangement if and only if nobody is paired with his spouse.

4.1.3 (a) If everyone who lost an eye also lost an arm, a leg and an ear, then there would be 70 people who lost all four.

(b) Let A be the set of people who lost an arm and L the set who lost a leg. How small can $A \cap L$ be? We have

$$|A \cap L| = |A| + |L| - |A \cup L| = 165 - |A \cup L| \geq 165 - 100 = 65.$$

We can now look at the set $D = A \cap L$ of double amputees and ask now many must have lost an eye. As above, we have

$$|D \cap I| = |D| + |I| - |D \cup I| \geq 65 + 70 - 100 = 35,$$

where I is the set of people who have lost an eye. Finally, we combine these people with the 75 who have lost an ear to conclude that at least $35 + 75 - 100 = 10$ must have lost all four. Thus $p \geq 10$. We can achieve this by insisting that everyone lost at least three things. If the people are numbered 1–100, we can do it as follows:

lost arm:	1–80
lost leg:	1–65 and 81–100
lost eye:	1–35 and 66–100
lost ear:	1–10 and 36–100

4.1.4. If a hand has some set of i properties, it means that the corresponding set of i suits is not present in the hand. Thus, such a hand is formed from a set of $(4-i)*13$ cards without repetition. This can be done in $\binom{52-13i}{k}$ ways. Thus $N_i = \binom{4}{i}\binom{52-13i}{k}$. By (4.3), the answer is

$$\binom{52}{k} - 4\binom{39}{k} + 6\binom{26}{k} - 4\binom{13}{k}.$$

4.1.5 (a) A number x has a factor in common with N if and only if it is divisible by one of the primes that divide N. Thus an element of $\underline{N}$ has no factor in common with N if and only if it is in none of the sets S_k.

(b) The intersection on the left side is the set of $x \in \underline{N}$ that are multiples of $b = p_{i_1} \cdots p_{i_r}$. These are b, $2b$, $3b$ …, $(N/b)b$. Thus the set has N/b elements, as was to be proved.

(c) By (4.3) and the previous result, we have

$$\varphi(N) \;=\; \sum_{I \subseteq \underline{n}} (-1)^{|I|} \frac{N}{\prod_{i\in I} p_i} \;=\; N \sum_{I\subseteq \underline{n}} \prod_{i \in I} \left(\frac{-1}{p_i}\right).$$

Replacing x_i by $-1/p_i$ in Example 1.13, we obtain the desired result.

4.1.6 (a) This can be done in various ways. One way is to permute the rows and columns of A so that the indices in K now appear as the top $|K|$ rows and leftmost $|K|$ columns. Everything must be zero except the $(n-|K|) \times (n-|K|)$ matrix in the lower right corner, which can be anything. This matrix contains $(n-|K|)^2$ entries each of which can be zero or one so the number of such matrices is $2^{(n-|K|)^2}$ by the Rule of Product.

(b) We use the Principle of Inclusion and Exclusion. Let S_i be the set of matrices with the ith row and ith column consisting entirely of zeroes. Then

$$|S_{i_1} \cap \cdots \cap S_{i_r}| \;=\; z(K) \quad \text{where} \quad K \;=\; \{i_1, \ldots, i_r\}.$$

Thus $N_r = \binom{n}{r} z_r$ and so the answer is

$$g_n \;=\; \sum_{r=0}^{n} (-1)^r \binom{n}{r} z_r \;=\; \sum_{r=0}^{n} (-1)^r \binom{n}{r} 2^{(n-r)^2}$$

4.1.7. Let S_i be those lists in which c_i is adjacent to c_i. Consider a list in $S_{i_1} \cap \cdots \cap S_{i_r}$. Using the hint, this can be thought of as a list made from $2m-r$ symbols, where for the present we regard the two occurrences of the symbol c_i as different Since the list is a rearrangement of the symbols, there are $(2m-r)!$ such lists. However, $m-r$ pairs of the symbols are identical and we have treated them as different. There are 2^{m-r} ways to treat such symbols as different. Thus $N_r = \binom{m}{r}(2m-r)!/2^{m-r}$.

4.1.8. There does not seem to be a simple formula. Let t be the number of S_i^3, p the number of S_j^2 for which S_j^3 is also required and p' the number of other S_j^2's. Then

$$N_r \;=\; \sum \binom{m}{t, p'} \binom{t}{p} (3m - 2t - p')!,$$

where the sum ranges over all values of t, p and p' such that $t + p + p' = r$.

4.1.9. The proof is practically the same as that given for Theorem 4.1. Instead of asking how much $s \in S$ contributes to the sums, ask how much $\Pr(s)$ contributes.

4.1.10. Suppose $s \in S$ lies in exactly j of the S_i. (Note that $j \le m$.) It contributes to exactly $\binom{j}{r}$ of the sum

$$N_r \;=\; \sum |S_{i_1} \cap \cdots \cap S_{i_r}|.$$

This is correct even when $r > j$ and $\binom{j}{r} = 0$. Thus s contributes

$$\sum_{i=0}^{m-k} (-1)^i \binom{k+i}{i}\binom{j}{k+i}.$$

to (4.16). We need to show that the sum is zero unless $j = k$. When $j < k$ all terms are zero. When $j = k$, only the $i = 0$ term is nonzero and it equals 1. Suppose $k < j \le m$. The terms in the sum are zero because of $\binom{j}{k+i}$ whenever $i > j - k$. Since $j \le m$, we can replace the upper limit of the sum with $j - k$. Note that

$$\binom{k+i}{i}\binom{j}{k+i} \;=\; \frac{(k+i)!}{i!\,k!}\frac{j!}{(k+i)!\,(j-k-i)!} \;=\; \frac{j!}{k!\,(j-k)!}\frac{(j-k)!}{i!(j-k-i)!} \;=\; \binom{j}{k}\binom{j-k}{i}.$$

Thus we can rewrite the sum as

$$\sum_{i=0}^{j-k}(-1)^i\binom{j}{k}\binom{j-k}{i} \;=\; \binom{j}{k}(1-1)^{j-k} = 0.$$

4.1.11 (a) Let the notation be as in the proof of the Principle of Inclusion and Exclusion. The proof given in the text is easily adjusted to prove s contributes exactly $c_{t-1}(X)$ to $\sum_{i=0}^{t-1}(-1)^i S_i$. Thus the sum will be a lower bound when t is even and an upper bound when t is odd. Including the term $(-1)^t S_t$ in the sum changes upper bounds to lower bounds and vice versa since we are now considering $c_t(X)$. By considering the cases of t even and t odd separately, it is easy to see that the inequalities follow.

(b) This can be proved by induction on t using $\binom{|X|}{t} = \binom{|X|-1}{t} + \binom{|X|-1}{t}$.

4.1.12. Since $|S| = N_0$, we have $|S_1 \cup \cdots \cup S_m| \;=\; N_0 - E$. Thus Bonferroni's inequalities give us

$$-N_t \;\le\; -|S_1 \cup \cdots \cup S_m| - \sum_{r=1}^{t-1}(-1)^r N_r \;\le\; N_t$$

and so

$$-N_t \;\le\; |S_1 \cup \cdots \cup S_m| - \sum_{r=1}^{t-1}(-1)^{r-1} N_r \;\le\; N_t.$$

4.1.13 (a) Let $m = 2$. Initially the N array contains

$$2: \quad N_2 \qquad 1: \quad N_1 \qquad 0: \quad N_0.$$

With $j = 0$, we do $i = 1$ and then $i = 0$. The N array now contains

$$2: \quad N_2 \qquad 1: \quad N_1 - N_2 \qquad 0: \quad N_0 - (N_1 - N_2).$$

With $j = 1$, we obtain

$$2: \quad N_2 \qquad 1: \quad (N_1 - N_2) - N2 \qquad 0: \quad N_0 - (N_1 - N_2).$$

Equation (4.16) gives

$$E_2 \;=\; N_2 \qquad E_1 \;=\; N_1 - 2N_2 \qquad E_0 \;=\; N_0 - N_1 + N_2,$$

which agrees with the values computed by the algorithm. You can carry out similar calculations for $m = 3$.

(b) This can be done by carefully carrying out the steps in the algorithm.

(c) After no iterations (that is, at the start of the algorithm), N_r contains s as many times as there is set of r indices for which (4.17) is true. If s appears in exactly p of the S_i, this number is $\binom{p}{r}$. We now use induction on t, having done the case $t = 0$. After $t-1$ iterations, formula (4.18) is true when t is replaced by anything smaller in it. In particular, it holds with t replaced by $t-1$.

We must now focus on the inner loop of the algorithm. What does it do? Since N_m never changes, neither does N_m^*. Formula (4.18) gives 0 or 1 for all t according as $p < m$ or $p = m$ ($p > m$ is impossible). This is the correct answer for both N_m and E_m.

Back to the action of the inner loop. Again we can prove it by induction, but now we are going from N_m^* down to N_0^*. We dealt with N_m^* in the previous paragraph. If the inner loop has done the correct thing with N_{r+1}^*, then the number of times s appears in the new version of N_r^* is $\mu(p,r,t-1) - \mu(p,r+1,t)$. There are various cases to consider. We'll just look at one, namely $\binom{p-(t-1)}{r-(t-1)} - \binom{p-t}{(r+1)-t}$. Using $\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}$, we have

$$\binom{p-(t-1)}{r-(t-1)} - \binom{p-t}{(r+1)-t} \;=\; \binom{p-t+1}{r-t+1} - \binom{p-t}{r-t+1} \;=\; \binom{p-t}{r-t},$$

which is what we needed to prove. We leave the other cases in (4.18) to you. The last sentence in the exercise follows from the fact that all the numbers we calculate are nonnegative. (This takes care of the problem of how we should interpret the multiset difference $A-B$ if s appears more often in B than it does in A.)

When $t \ge m$, the only time the binomial coefficient is used in (4.18) is when $t = p = m$ and it then has the value $\binom{0}{r-m}$, which is zero unless $r = m$, when it is 1. Thus, for $t \ge m$, $\mu(p,r,t)$ equals 1 if $r = p$ and 0 otherwise. Hence N_r^* is a set containing precisely those elements that are in exactly r of the S_i.

(d) This is implicit in the proof for (c).

4.1.14 (a) S_i is the set of permutations of n that fix i and so

$$|S_{i_1} \cap \ldots \cap S_{i_k}| \;=\; (n-k)!.$$

This leads to

$$D_n(k) \;=\; \sum_{i=0}^{n-k} (-1)^i \binom{k+i}{i} \binom{n}{k+i} (n-k-i)!.$$

(b) Choose k fixed points AND derange the remaining $n-k$ points.

(c) We want to prove that

$$\sum_{i=0}^{n-k} (-1)^i \binom{k+i}{i} \binom{n}{k+i} (n-k-i)! \;=\; \binom{n}{k} (n-k)! \sum_{i=0}^{n-k} \frac{(-1)^i}{i!}.$$

It suffices to observe that $\binom{k+i}{i}\binom{n}{k+i}(n-k-i)! \;=\; \frac{(k+i)!}{k!\,i!}\frac{n!}{(k+i)!\,(n-k-i)!}(n-k-i)! \;=\; \frac{n!}{k!\,i!}$ and $\binom{n}{k}\frac{(n-k)!}{i!} \;=\; \frac{n!}{k!\,i!}$.

4.1.15. Let S_i be the subset of $\underline{n}$ divisible by a_i. Then, N_t is the sum over all t-subsets T of A of $\lfloor n/\text{lcm}(T)\rfloor$, where $\text{lcm}(T)$ is the least common multiple of the elements of T and the floor $\lfloor x\rfloor$ is the largest integer not exceeding x. For the various parts of the exercise you need the following.

- If all elements of A divide n, then $\lfloor n/\text{lcm}(T)\rfloor = n/\text{lcm}(T)$.
- If no two elements of A have a common factor, then $\text{lcm}(T) = \prod_{i\in T} i$.

(a) Using the previous comments in the special case $k = 0$, we obtain after some algebra

$$\sum_{i=0}^{m}(-1)^i N_i \;=\; n\prod_{a\in A}\left(1-\frac{1}{a}\right),$$

which is the Euler phi function when A is the set of prime divisors of n.

(b) The comment for (a) applies in this case as well.

(c) There is no simple formula even when $k = 0$ because the floor function cannot be eliminated.

(d) Now we cannot even eliminate the lcm function.

4.1.16. We cannot have x less than x, which is required by (P-1).

4.1.17. In all cases, what we must do is prove that (P-1), (P-2) and (P-3) hold. We omit most of them.

(d) Since $x/x = 1$, (P-1) is true. Suppose that $x\rho y$ and $y\rho x$. Then x/y and y/x are both integers. Since $(x/y)(y/x) = 1$, the only possible integer values for x/y and y/x are ± 1. Since x and y are positive, it follows that $x/y = 1$ and so (P-2) is true. Suppose that x/y and y/z are integers. Then so is x/z and so (P-3) is true.

4.1.18. Since $x\rho x$, we have $x\tau x$ and so (P-1) is true for (S,τ). Suppose $x\tau y$ and $y\tau x$. Then $y\rho x$ and $x\rho y$. By (P-2) for the poset (S,ρ), $x = y$ and so (P-2) is true for (S,τ). Suppose $x\tau y$ and $y\tau z$. Then $z\rho y$ and $y\rho x$ and so $z\rho x$. Thus $x\tau z$ and so (P-3) is true.

4.1.19. Since every set is the union of itself, $x\rho x$. Suppose $x\rho y$ and $y\rho x$. Let b_y be a block of y. Since $x\rho y$, $b_x \subseteq b_y$ for some block b_x of x. Since $y\rho x$, $b'_y \subseteq b_x$ for some block b'_y of y. Since blocks of a partition are either equal or disjoint and since $b'_y \subseteq b_x \subseteq b_y$, we have $b'_y = b_y$ and so $b_x = b_y$. This proves that every block of y is a block of x. Hence $x = y$ and so (P-2) is true. It is easy to prove (P-3).

4.1.20. The proofs of (P-1), (P-2) and (P-3) are all straightforward uses of "and." We do (P-3).

$$\begin{aligned}
(x,x')\pi(y,y') \text{ and } (y,y')\pi(z,z') \quad &\text{means} \quad \big(x\rho y \text{ and } x'\tau y'\big) \text{ and } \big(y\rho z \text{ and } y'\tau z'\big);\\
&\text{which is} \quad \big(x\rho y \text{ and } y\rho z\big) \text{ and } \big(x'\tau y' \text{ and } y'\tau z'\big);\\
&\text{which implies} \quad x\rho z \text{ and } x'\tau z' \qquad\qquad \text{by (P-3) for } \rho \text{ and } \tau;\\
&\text{which means} \quad (x,x')\pi(z,z').
\end{aligned}$$

4.1.21 (a) With each element $s \in S$, associate a set $g(s)$ such that $s \in S_i$ if and only if $i \in g(s)$. Then E_k counts those $s \in S$ for which $|g(s)| = k$. Since the number of $s \in S$ with $g(s) = y$ is $e(y)$, the sum of $e(y)$ over $|y| = k$ also counts those s.

(b) An element s is counted in (4.14) if and only if it belongs to all S_i for which $i \in x$. This is the same as the definition of the set intersection.

(c) The sum of $e(x)$ over all x of since k is E_k. Putting this together with (4.15), we have

$$E_k = \sum_{|x|=k} \sum_{y \supseteq x} (-1)^{|y|-k} f(y) = \sum_{|y|\ge k} \sum_{\substack{x \subseteq y \\ |x|=k}} (-1)^{|y|-k} f(y) = \sum_{|y|\ge k} \binom{|y|}{k} (-1)^{|y|-k} f(y).$$

The sum of $f(y)$ in (b) over all y of size t is N_t. Collecting terms according to $|y|$, we have

$$E_k = \sum_{t=k}^{m} \binom{t}{k} (-1)^{t-k} N_t = \sum_{i=0}^{m-k} \binom{i+k}{k} (-1)^i N_{k+i},$$

where we set $t = i + k$. Now use $\binom{i+k}{k} = \binom{k+i}{i}$.

Section 4.2

4.2.1. The number of 6-long sequences made with B, R and W is $3^6 = 729$, which is much too long. The number of 6-long sequences in which adjacent beads differ in color is $3 \times 2^5 = 96$, which is more manageable, but still quite long. We won't list them. We could "cheat" by being a bit less mechanical: If the necklace contains a B, we could start with it. There are $2^5 = 32$ such necklaces, a manageable number. The only necklace without B must alternate R and W, so there is only one of them. Here are the 32 other necklaces, where a number preceding a necklace is the first place it appears in the list when considered circularly or flipped over. A zero means it was rejected because the first and last beads are the same.

1: BRBRBR	2: BRBRBW	0: BRBRWB	3: BRBRWR	2: BRBWBR	4: BRBWBW
0: BRBWRB	5: BRBWRW	0: BRWBRB	6: BRWBRW	0: BRWBWB	7: BRWBWR
3: BRWRBR	8: BRWRBW	0: BRWRWB	9: BRWRWR	2: BWBRBR	4: BWBRBW
0: BWBRWB	8: BWBRWR	4: BWBWBR	10: BWBWBW	0: BWBWRB	11: BWBWRW
0: BWRBRB	7: BWRBRW	0: BWRBWB	7: BWRBWR	5: BWRWBR	11: BWRWBW
0: BWRWRB	12: BWRWRW				

4.2.2. There are 8 solutions with equal numbers of B's and W's, 5 with 5 B's, 4 with 6 B's and one each with 7 and 8 B's. this gives us a total of

$$8 + 2(5 + 4 + 1 + 1) = 30.$$

4.2.3 (a) Since 4 beads are used, at most 4 different kinds of beads are used. We can construct an arrangement of beads by choosing the number of types that must appear (1, 2, 3 OR 4), choosing that many types of beads from the r types AND then choosing an arrangement using all of the types of beads that we chose.

(b) Trivially, $f(1) = 1$. For $f(2)$, our decision will be the number of beads of the first type that appear. After that, it is easy. This gives us $1 + 2 + 1 = 4$. For $f(3)$, our decision will be which bead appears twice. This gives us $3 \times 2 = 6$ For $f(4)$, each bead appears once and there are 3 possibilities. Thus

$$F(r) = \binom{r}{1} + \binom{r}{2}4 + \binom{r}{3}6 + \binom{r}{4}3,$$

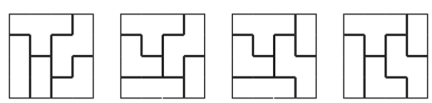
which can be rewritten as $r(r+1)(r^2 + r^2)/8$, if desired.

4.2.4. This has nothing at all to do with symmetries. If there are $F(r)$ ways to use r types of "beads" to do something and $f(r)$ ways to do it so that each bead is used at least once, then $F(r) = \sum_{k=0}^{r} \binom{r}{k} f(k)$. If at most M beads are used, then the upper limit on the sum can be replaced by M because (i) $f(k) = 0$ for $k > M$ and (ii) $\binom{r}{j} = 0$ for any j satisfying $r < j \le M$.

4.2.5. The problem can be solved by either decision tree method. It is useful to note that all solutions must begin with h because any board that starts with v can be flipped about a NW-SE (135°) diagonal to give one that starts with h. Also note that a lexically least sequence that starts with hv determines the entire sequence. (To see this, note that it starts hvv and look at rotations of the board.)

We will use the second method. Our first decision will be the number of entire rows and/or columns that are covered by two whole dominoes. For example, two dominoes in the top row or two dominoes in the third column. Note that we cannot simultaneously cover a row and a column because they overlap. Let the number be L. The possible values of L are 0, 1, 2 and 4. (You should find it easy to see why $L = 3$ is impossible.) Note that we can always use the symmetries to make the first domino horizontal. For $L = 4$, there is obviously only one solution and its lex minimal form is *hhhhhhhh*. For $L = 0$, we use Method 1 to obtain *hvvhvvhh* as the the only solution. (Beware: reading the sequence in reverse does not correspond to a symmetry of the board.) For $L = 1$, we note that the entire row or column must be at the edge of the board. Suppose it is the first row. Refer back to Figure 3.15 to see that the only way to complete the board without increasing L is *hvvvvh*. This is already lex minimal: *hhhvvvvh*. Suppose $L = 2$. By rotation, we can assume we have two full rows and, because they cannot be in the middle, one of them is the first row. Again, refer to Figure 3.15 to find how many ways we can complete the board with one more horizontal row. This leads to six solutions: *hhhhhvvh*, *hhhvvhhh*, *hhhhvhvh*, *hhvhvhhh*, *hhhhvvvv* and *hhvvvvhh*. This gives a total of nine solutions.

4.2.6. There are 4 solutions:



4.2.7. When we write out our answers, they will be in the form suggested in the problem, without the surrounding boxes. To obtain the lex least solutions, we must linearly order the faces. Our order will be the line of four side faces from left to right, then the top and, finally, the bottom. We use B, R and W to denote the colors. and b, r and w to denote the number of faces of each color.

(a) Our first decision will be the number of black faces. By interchanging black and white, a solution with b black faces can be converted to one with $6-b$, so we only need look at $b = 0$ 1, 2 and 3. For $b = 0$ and $b = 1$, there are obviously only one solution. For $b = 2$, we must decide whether to put the second black face adjacent or opposite the first one. Here are the 4 solutions for $b < 3$.

W	W	W	W
W W W W	B W W W	B B W W	B W B W
W	W	W	W

For $b = 3$, our second decision is whether or not all three black faces share a common vertex. This leads to just 2 solutions:

B	W
B B W W	B B B W
W	W

1,1,4	W B R W W W	W B W R W W	
1,2,3	W B R R W W	R B R W W W	W B R W R W
2,2,2	R B B W W R	R B R B W W	W B R B R W
2,2,2	W B B R R W	R B B R W W	W B B R W R

Figure S.4.1 The distinct painted cubes with various numbers of faces painted Black, Red and White.

Doubling the answers for $b < 3$ to get those for $b > 3$ gives us 10 solutions.

(b) In the previous solution, we can limit ourselves to $b \le 3$. When $b = 3$, we need to check whether or not one solution is converted to the other when black and white are interchanged. They are not, so $b = 3$ still gives 2 solutions for a total of 6.

(c) The mirror image of each of the 10 solutions is equivalent to itself, so there are still 10 solutions.

(d) Our first decision will be the list b, r, w. By interchanging colors, we need only consider the situations where $b \le r \le w$. This gives us 1,1,4, 1,2,3 and 2,2,2. Interchanging colors in all possible ways gives rise to 3, 6 and 1 solutions, respectively, for each solution found. For 1,1,4, our decision will be whether B and R are on adjacent or opposite faces. Each leads to one coloring. For 1,2,3 our first decision will be the number of R's that are adjacent to the B. One adjacency gives 1 solution and two give 2 solutions, depending on whether the R's are adjacent or opposite each other. For 2,2,2, our first decision will be whether or not the B's are adjacent or opposite. Our second decision will be whether or not the R's are adjacent or opposite. Each choice leads to 1 solution except when the B's are adjacent and the R's are adjacent. In this case there are more solutions. One possibility is to have the 4 sides be BBRR. Another possiblility is to have the 4 sides be BBRW and then place the additional R on either the top or the bottom. These last two possibilities are mirror images of each other, but we cannot transform one to the other with just rotations. The solutions are given in Figure S.4.1. This gives us $2 \times 3 + 3 \times 6 + 6 = 30$ solutions.

(e) If all 3 colors appear, there are 30 solutions. If only 1 color appears, there are obviously 3 solutions. What if exactly 2 colors appear, we can first choose the 2 colors AND then use them. By the first part of this exercise, there are $10 - 2 = 8$ ways to use the colors so that both appear. Thus we have $30 + 3 + \binom{3}{2}8 = 57$ solutions.

(f) Note that no color can appear more than 3 times on any given cube. Also note that at most 6 colors appear on any given cube. By looking over our previous work, we find, in the notation of Exercise 4.2.4, that $f(0) = f(1) = 0$, $f(2) = 1$ and $f(3) = 8$. By looking at decision trees for the color counts 1,1,1,3 and 1,1,2,2, we find that $f(4) = \binom{4}{1}2 + \binom{4}{2}5 = 32$. Consider $f(5)$ which has just the one color count list 1,1,1,1,2. There is one way to place the repeated colors. The partially colored cube can be transformed into itself be leaving it fixed or by rotating it so that

z	0	1	2	2	2	3	3	3	4	4	4	4	4
D			1	2	3	2	3	3	2	3	3	3	3
A						2	0	1	2	3	3	3	3
000	1	0	0	0	0	0	0	0	0	0	0	0	0
001	1	1	0	1	1	0	1	0	0	0	0	0	0
010	1	1	1	1	1	0	1	1	0	1	0	0	0
100	1	1	1	1	1	1	1	1	1	1	1	1	0
011	1	1	1	0	1	1	0	1	0	1	1	1	1
101	1	1	1	1	1	1	0	1	1	1	0	1	1
110	1	1	1	1	1	1	1	0	1	0	1	1	1
111	1	1	1	1	0	1	1	1	1	0	1	0	1

Figure S.4.2 The 13 different Boolean functions for $n = 3$. Each column gives a function having the values of z, D and A at the head of the column. The row label describes the argument; e.g., an entry in row 101 is $f(1, 0, 1)$.

the two colored faces are interchanged. This means that whenever we color the remaining 4 faces with 4 *distinct* colors, there will be exactly one other coloring that is equivalent to it. Thus $f(5) = \binom{5}{1}(4!/2) = 60$. If you experiment a bit, you will discover that there are 24 symmetries of the cube. If all the faces are colored differently, each of the symmetries leads to an equivalent coloring that looks different. Thus $f(6) = 6!/24 = 30$. Putting all this together, we have

$$F(r) = \binom{r}{2} + \binom{r}{3}8 + \binom{r}{4}32 + \binom{r}{5}60 + \binom{r}{6}30.$$

4.2.8 (a) We can describe the vertices of a square or cube by specifying their coordinates. For the square in the exercise, the coordinates are $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$. We can interpret a the digit d at the corner with coordinates (x, y) as saying that $f(x, y) = d$. In a similar manner, a cube corresponds to $n = 3$.

(b) We'll just do the cube. Permutions of the arguments correspond to symmetries that do not move the point $(0, 0, 0)$. Replacing x_i with $x_i \oplus 1$ corresponds to reflection in the plane $x_i = 1/2$. Thus everything except the $c\oplus$ part is explained in terms of rotations and reflections of the cube. Conversely, given any symmetry of the cube, it can be interpreted in this manner. For example, if x-axis is mapped so that it is parallel to the z-axis, then $\sigma(3) = 1$. The image of $(0, 0, 0)$ is (d_1, d_2, d_3). Finally, we note that $c = 1$ corresponds to interchanging the values of zero and one assigned to the vertices of the cube.

(c) Our first decision the value for z, the number of corners of the cube with zeroes. Our second decision, when neeeded was the dimension D of the part of the cube that contained all the zeroes; e.g., if they were all on one face, $D = 2$. Given any vertex v with a zero, we can ask how many zeroes we can reach from v by going to the other end of an edge containing v. (This number is 0, 1, 2 or 3.) If a third decision was needed, it was A, the maximum of this number over all such v. We then used Method 1. A table of the resulting 13 functions is given in Figure S.4.2.

Section 4.3

4.3.1. The image of F is all k element subsets of $\underline{n}$. $F^{-1}(x)$ consists of all possible ways to arrange the elements of x in a list. Since we are able to count lists, we know that there are $k!$ such arrangements. We also know that $|A| = n!/(n-k)!$. Thus the coimage of F consists of $C(n,k)$ blocks all of size $k!$ and the union of these blocks has $n!/(n-k)!$ elements. Thus $C(n,k) = \frac{n!}{k!(n-k)!}$.

4.3.2. This is a problem for Chapter 1 since it deals with circular sequences of distinct things! An n-long circular sequence of *distinct* can be cut in n places to get n different n-long lists and each list is obtained exactly once this way. Thus the answer is $k(k-1)\cdots(k-n+1)/n$.

4.3.3. Note that $N(\gamma) = 0$ unless $\gamma \in P_8$ or $\gamma \in P_5$. In the former case, $N(\gamma) = \binom{8}{3} = 56$ and in the latter case, $N(\gamma) = \binom{2}{1}\binom{3}{1} = 6$. Thus there are $(56 + 4 \times 6)/16 = 5$ necklaces.

4.3.4. For $\gamma \in P_8$, $N(\gamma) = k^8$. For $\gamma \in P_5$, $N(\gamma) = k^5$. For $\gamma \in P_4$, $N(\gamma) = k^4$. For $\gamma \in P_2$, $N(\gamma) = k^2$. For $\gamma \in P_1$, $N(\gamma) = k$. Thus the answer is

$$\tfrac{1}{16}(k^8 + 4k^5 + 5k^4 + 2k^2 + 4k).$$

4.3.5 (a) The second line consists of the first line circularly shifted by c, an integer between 0 and $n-1$; i.e., the second line is $s_1, s_2, \ldots, s_n$, where $s_t = c + t$ if this is at most n and $c + t - n$, otherwise.

(b) In addition to the elements of the cyclic group, we have permutations whose second lines are cyclic shifts of $n, \ldots, 2, 1$.

(c) There are 0, 1 or 2 cycles of length 1 and the remaining cycles are all of length 2. If n is odd, there is always exactly one cycle of length 1. If n is even, there is never exactly one cycle of length 1. You can write down the cycles as follows. All numbers that are mentioned are understood to have an appropriate multiple of n added to (or subtracted from) them so that they lie between 1 and n inclusive. If n is odd, choose a cycle (k). The remaining cycles are $(k-t, k+t)$ where $1 \le t < n/2$. If n is even, choose $k \le n/2$. There are two ways to proceed. First, we could have all cycles of the form $(k-t+1, k+t)$ where $1 \le t \le n/2$. Second, we could have (k), $(k+n/2)$ and all cycles of the form $(k-t, k+t)$ where $1 \le t < n/2$.

4.3.6 (a) Number the squares 1 to 16, starting in the upper left corner and proceeding left to right one row at a time. There are just four permutations of the board, namely the cyclic group on 4 things. Here's what the permutations other than e do the the squares of the board:

$$(1,4,16,13)(2,8,15,9)(3,12,14,5)(6,7,11,10)$$

$$(1,13,16,4)(2,9,15,8)(3,5,14,12)(6,10,11,7)$$

$$(1,16)(4,13)(2,15)(8,9)(3,14)(5,12)(6,11)(7,10).$$

Thus the number of ways to choose 8 squares is

$$\tfrac{1}{4}\left(\binom{16}{8} + 2\binom{4}{2} + \binom{8}{4}\right).$$

(b) We now have the dihedral group. The 4 additional permutations are

$$(1)(2,5)(3,9)(4,13)(6)(7,10)(8,14)(11)(12,15)(16)$$

$$(1,16)(2,12)(3,8)(4)(5,15)(6,11)(7)(9,14)(10)(13)$$

$$(1,4)(2,3)(5,8)(6,7)(9,12)(10,11)(13,16)(14,15)$$

$$(1,13)(2,14)(3,15)(4,16)(5,9)(6,10)(7,11)(8,12).$$

Thus the number of ways to choose 8 squares is

$$\tfrac{1}{8}\Big(\tbinom{16}{8}+2\tbinom{4}{2}+\tbinom{8}{4}2\big[\tbinom{4}{4}\tbinom{6}{2}+\tbinom{4}{2}\tbinom{6}{3}+\tbinom{4}{0}\tbinom{6}{4}\big]+2\tbinom{8}{4}\Big).$$

4.3.7. The proof in the text shows that the right side of the given equality is $|G|\sum_{g\in G}N(g)$. By (4.20), the left side is

$$\sum_{y\in S}|I_x| \;=\; |G|\sum_{y\in S}\frac{1}{|B_y|}.$$

The rest of the proof follows easily by adapting what was done in the text. This seems to be a shorter proof than the one in the text. Why didn't we use it? First, it's not particularly shorter; however, it is a bit cleaner. Unfortunately, it requires starting with the completely unmotivated double summation in which we have interchanged the order of the sums.

4.3.8 (a) We don't list the coverings. In general, the coverings can be made by stringing together two kinds of "beads:" a single vertical domino and a pair of horizontal dominoes, one above the other.

(b) In the previous result, replace the vertical domino beads by ones and the horizontal pairs by twos. Since a vertical covers one column and a pair of horizontals covers two, the numbers add up to n.

(c) A sequence of ones and twos completely determines a board. Symmetries of the board either leave a sequence unchanged or reverse its order.

(d) The group of symmetries of the sequences of ones and twos contains just two elements: the identity e and the reversal of left and right, say g. Obviously, $N(e)=D(n)$. $N(g)$ is associated with the number of ways to cover a board of roughly length $n/2$. If n is odd, g leaves the middle column of the board fixed and interchanges columns j and $n+1-j$ for $1\le j<n/2$. If the board is to be unchanged by g, the middle column must contain a vertical domino, the first $(n-1)/2$ columns can contain any covering, and the last $(n-1)/2$ must contain their image under g. Thus $N(g)=D((n-1)/2)$ when n is odd. When n is even, a similar argument can be used. Now there is no center column. We can either cover the first $n/2$ columns or cover the first $n/2-1$ columns and place a pair of horizontal dominoes covering columns $n/2$ and $n/2+1$.

Section 5.1

5.1.1. The sum is the number of ends of edges since, if x and y are the ends of an edge, the edge contributes 1 to the value of $d(x)$ and 1 to the value of $d(y)$. Since each edge has two ends, the sum is twice the number of edges.

5.1.2. To specify a graph we must choose $E\in\mathcal{P}_2(V)$. Let $N=|\mathcal{P}_2(V)|$. Then there are 2^N possible subsets E of $\mathcal{P}_2(V)$ and $\binom{N}{q}$ of them have cardinality q. Since $|\mathcal{P}_2(V)|=\binom{n}{2}$, we are done with (a) and (b). Since there are 2^N graphs, each has probability 2^{-N} and so the probability in (c) is $2^{-N}\binom{N}{q}$.

5.1.3. The graph with

$$\varphi \;=\; \begin{pmatrix} a & b & c & d & e & f & g & h & i & j & k\\ C & C & F & A & H & E & E & A & D & A & A\\ C & G & G & H & H & H & F & H & G & D & F \end{pmatrix}.$$

is isomorphic to Q. The correspondence between vertices is given by

$$\begin{pmatrix} A & B & C & D & E & F & G & H \\ H & A & C & E & F & D & G & B \end{pmatrix}$$

where the top row corresponds to the vertices of Q. The graph with

$$E \;=\; \{1,2,3,4,5,6,7,8,9,10,11\} \quad \text{and} \quad \varphi \;=\; \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ A & E & E & E & F & G & H & B & C & D & E \\ G & H & E & F & G & H & B & C & D & D & H \end{pmatrix}.$$

is not ismorphic to Q. One edge needs to be deleted from $P'(Q)$ and one added.

5.1.4. If a pictorial representation of R can be created by labeling $P'(Q)$ with the edges and vertices of R, then R has degree sequence $(0,2,2,3,4,4,4,5)$. The converse is false (find a counterexample).

5.1.5 (a) There is no graph Q with degree sequence $(1,1,2,3,3,5)$ since the sum of the degrees is odd.

(b) There are such a graph. You should draw an example.

(c) Up to labeling, the graph is unique. Take $V = \{1,\ldots,6\}$ and

$$E \;=\; \big\{\{1,6\},\ \{2,6\},\ \{2,4\},\ \{3,6\},\ \{3,5\},\ \{4,6\},\ \{4,5\},\ \{5,6\}\big\}$$

(d) A graph with degree sequence $(3,3,3,3)$ has $(3+3+3+3)/2 = 6$ edges and, of course 4 vertices. That is the maximum $\binom{4}{2}$ of edges that a graph with 4 vertices can have. It is easy to construct such a graph. This graph is called the *complete* graph on 4 vertices.

(f) There is no simple graph (or graph without loops or parallel edges) with degree sequence $(3,3,3,5)$.

(g) Similar arguments to the $(3,3,3,3)$ case apply to the complete graph with degree sequence $(4,4,4,4,4)$.

Section 5.2

5.2.1. Let ν and ε be the bijections.

(a) This follows from the fact that ν and ε are bijections.

(b) This can be seen intuitively from the drawing of the unlabeled graph. If you want a more formal proof, first note that the degree of a vertex v is the number of edges e such that $v \in \varphi(e)$. Now use the fact that $v \in \varphi(e)$ is equivalent to $\nu(v) \in \varphi'(\varepsilon(e))$.

5.2.2. Each of (a) and (c) has just one pair of edges with the same endpoints, while (b) and (d) each have two pairs. Thus neither (b) nor (d) is equivalent to (a) or (c). Vertex 1 of (b) has degree 4, but (d) has no vertices of degree 4. Thus (b) and (d) are not equivalent. It turns out that (a) and (c) are equivalent. We leave it to you to find ν and ε.

5.2.3 (a) This is exactly like the next problem with the transpose, ^{t}, replaced by inverse, $^{-1}$, everywhere.

(b) Let I be the $n \times n$ identity matrix. Since $A = IAI^t$, $A \simeq A$. Suppose that $A \simeq B$. Then $B = PAP^t$ for some nonsingular P. Multiplying on the left by P^{-1} and on the right by $(P^{-1})^t = (P^t)^{-1}$, we have

$$(P^{-1})B(P^{-1})^t \;=\; (P^{-1}P)A(P^t(P^{-1})^t) \;=\; (P^{-1}P)A(P^t(P^t)^{-1}) = A.$$

Thus $B \simeq A$. Suppose that $A \simeq B \simeq C$. Then we have nonsingular P and Q such that $B = PAP^t$ and $C = QBQ^t$. Thus $C = Q(PAP^t)Q^t = (QP)A(P^tQ^t) = (QP)A(QP)^t$. This proves transitivity.

5.2.4. We will just explain those that are not equivalence relations.

(c) We could have three students, Alice, Bill and Chris, with Chris having a class with Alice and another class with Bill, but Alice and Bill have no classes in common. This would make Chris equivalent to both Alice and Bill, but Alice and Bill would not be equivalent. This violates the transitive property of equivalence relations.

(d) Consider the numbers 0, 0.0008, 0.0016. As in the previous question, the transitive property fails.

5.2.5. Let $E \in \mathcal{P}_2(V)$ and $E' \in \mathcal{P}_2(V')$. Write $G = (V, E) \simeq (V', E') = G'$ if and only if there is a bijection $\nu: V \to V'$ such that $\{u, v\} \in E$ if and only if $\{\nu(u), \nu(v)\} \in V'$.

We could show that this is an equivalence relation by adapting the proof in Example 5.5. An alternative is to show how this definition leads to the equivalence relation for G and G' interpreted as graphs. We'll take this approach. In this case φ and φ' are identity maps. Define $\varepsilon(\{u, v\}) = \{\nu(u), \nu(v)\}$. By our definition in the previous paragraph, $\varepsilon: E \to E'$ is a bijection. Since φ and φ' are the identity, the requirement that $\varphi'(\varepsilon(e)) = \nu(\varphi(e))$ in the definition of graph isomorphism is satisfied.

5.2.6 (a) With π the identity, we see that $f \simeq f$. Suppose that $f(x) = g(\pi(x))$ for all $x \in A$ and let $y \in A$. Since π is a bijection, $\pi(x) = y$ for some $x \in A$. Thus $x = \pi^{-1}(y)$ and

$$g(y) \ = \ g(\pi(x)) \ = \ f(x) \ = \ f(\pi^{-1}(y)).$$

Thus the permutation π^{-1} proves that $g \simeq f$. (For those more familiar with manipulating functions, we could simply say: Since $f = g\pi$ and π is a bijection, we have $g = g\pi\pi^{-1} = f\pi^{-1}$.) Suppose that $f \simeq g \simeq h$ and that the permutations involved are π and σ. Then $f(x) = g(\pi(x)) = h(\sigma(\pi(x))$. Since $\sigma(\pi(x))$ is a permutation of A, $f \simeq h$.

(b) Call two functions $f, g: A \to B$ equivalent if there is permutation π of B such that $f = \pi g$. With π the identity function, we see that $f \simeq f$. Suppose that $f = \pi g$, then $g = \pi^{-1} f$ and so $g \simeq f$. Suppose that $f = \pi g$ and $g = \sigma h$. Then $f = (\pi\sigma)h$ and so $f \simeq h$.

(c) Suppose $f, g: A \to B$ and that there are permutations α and β of A and B respectively such that $f = \beta g \alpha$. Then call f and g equivalent. Using the identity permutations, we have that $f \simeq f$. Since $g = \beta^{-1} f \alpha^{-1}$, $g \simeq h$. Suppose that $g = \beta' h \alpha'$. Then $f = \beta(\beta' h \alpha')\alpha = (\beta\beta')h(\alpha'\alpha)$, and so $f \simeq h$.

5.2.7. The table is shown in Figure S.5.1. The entries which are 1 follow when you realize what is being counted. The LL row corresponds to ordered samples and the UL row to unordered samples, which have been considered in Chapter 1. The UL-surjection entry comes from the realization that our sample allows repetition but must include every element in b so that we are only free to choose $a - b$ additional elements. In the LU row, the fact that the range is unlabeled means that we can only distinguish functions that have different coimages. The UU row is associated with partitions of numbers. We use $p(n, k)$ to denote the number of partitions of n having exactly k parts.

A	B	all	injections	surjections
L	L	b^a	$b(b-1)\cdots(b-a+1)$	$b!S(a,b)$
L	U	$\sum_{k\le b} S(a,k)$	1	$S(a,b)$
U	L	$\binom{a+b-1}{a}$	$\binom{b}{a}$	$\binom{a-1}{a-b}$
U	U	$\sum_{k\le b} p(a,k)$	1	$p(a,b)$

Figure S.5.1 Some basic enumeration problems.

Section 5.3

5.3.1. Since $E \subseteq \mathcal{P}_2(V)$, we have a simple graph. Regardless of whether you are in set C or S, following an edge takes you into the other set. Thus, following a path with an odd number of edges takes you to the opposite set from where you started while a path with an even number of edges takes you back to your starting set. Since a cycle returns to its starting vertex, it obviously returns to its starting set.

5.3.2 (a) and (b) The solution to Exercise 5.3.1 shows that the graph is bipartite and the argument given there for even length cycles can be used practically as is for bipartite graphs: just replace C and V with A and B.

(c) Start sets A and B empty. Choose any vertex v_0 and place it in the set A. While there is some edge $\{u,v\}$ with $u \in A \cup B$ and $v \notin A \cup B$, place v in the set *not* containing u.

Since the graph is connected, all vertices will eventually be in A or B, so we produce a partition of V. Suppose that there is some edge $e = \{x,y\}$ with both ends in A. By the way A and B were constructed, there is a path from x to v_0 consisting solely of edges that were used to add vertices to A and B. If the vertices on the path are $c_0 = x, c_1, \ldots, c_n = v_0$, then $c_{2i} \in A$ and $c_{2i+1} \in B$. There is a similar path $d_0 = y, d_1, \ldots, d_m = v_0$. Let $c_i = d_j$ be the first common vertex on the two paths. It follows that $x, c_0, c_1, \ldots, c_i = d_j, d_{j-1}, \ldots, d_0 = y, x$ is a cycle of length $i+j+1$. Since c_i and d_j are the same vertex, they are both in A or both in B. Thus i and j are both even or both odd. Consequently $i+j+1$ is odd, contradicting the requirement that all cycle length be even. If the ends of e are in B a similar proof works.

(d) Whenever the previous algorithm stops and vertices remain, choose a remaining vertex and place it in A. Then continue with finding and edge e.

(e) The set in which the vertex y of edge e is placed by the algorithm is forced. We have a free choice when we arbitrarily select a vertex and place it in A. Thereafter, the algorithm places all the other vertices that lie in the same component and, as just noted, these placements into A or B are forced. In other words, we get one free choice of A or B for each component.

(f) We have already shown that all cycles in a bipartite graph have even length, so we must do the reverse. Let G be an arbitrary graph with no odd length cycles. Apply the algorithm for partitioning V into A and B. Now look back at the proof that the algorithm worked. We showed that if there was an edge with both ends in A or both in B then there was a cycle of odd length. Since G has no cycles of odd length, it follows that our algorithm provides the partitioning $V = A \cup B$ required in the definition of a bipartite graph.

5.3.3 (a) Let $e = \{u,v\}$ and let $f = \{v,w\}$ be the other edge. Since G is simple, $u \neq w$. Since e is a cut edge, u and v are in separate components of $(V, E - \{e\})$. Thus so are u and w. Since the graph induced by $V - \{v\}$ is a subgraph of $(V, E - \{e\})$, u and w are in separate components of it as well.

(b) Take two triangles and identify their tops. The merged top is a cut vertex but the graph has no isthmus.

(c) We will prove that e is a cut edge if and only if its ends u and v, say, lie in different components of $G' = (V, E - \{e\})$. The result will then follow because, first, if C is a cycle containing e, removal of e does not leave its ends in different components, and, second, if u and v are in the same components of G', then there is a path P connecting them in G' and P and e form a cycle in G.

Now back to the original claim. If u and v are in different components of G', then e is a cut edge. Suppose e is a cut edge of G. Since G is connected and every path in G that is not a path in G' contains e, it follows that if x and y are in different components of G' any path connecting them in G contains e. Let P be such a path and let u be the end of e first reached on P when starting from x. It follows that x and u are in one component of G' and that y and v (the other end of e) are one component, too. Since x and y are in different components, so are u and v.

(d) We claim that $v \in V$ is a cut vertex of G if and only if there are two edges e and e' both containing v such that no cycle of G contains both e and e'.

Proof. Suppose that v is a cut vertex. Let x and y belong to different components of the graph G'' induced by $V - \{v\}$. Any path from x to y in G must include v. Let P be such a path and let e and e' be the two edges in P that contain v. If e and e' were on a cycle C in G, then we could remove e and e' from P and add on $C - \{e, e'\}$ to obtain a route from x to y that does not go through v. Since this contradicts the fact that x and y are in different components of G", it follows that e and e' do not lie in a cycle.

The steps can be reversed to prove that if e and e' are edges incident with v that do not lie on a cycle, then v is a cut vertex: Let x and y be the other vertices on e and e'. Since e and e' do not lie on a cycle, every path from x to y must include either e or e' (or both), and hence includes v. Since there is no path from x to y not including v, they are in different components of G".

5.3.4. The definitions of connected graphs and trees would result in the same structures.

5.3.5 (a) The graph is not Eulerian. The longest trail has 5 edges, the longest circuit has 4 edges.

(b) The longest trail has 9 edges, the longest circuit has 8 edges.

(c) The longest trail has 13 edges (an Eulerian trail starting at C and ending at D). The longest circuit has 12 edges.

(d) This graph has an Eulerian circuit (12 edges).

5.3.6 (a) The graph is Hamiltonian.

(b) The graph is Hamiltonian.

(c) The graph is not Hamiltonian. There is a cycle that includes all vertices except K.

(d) The graph is Hamiltonian.

Section 5.4

5.4.1. We first prove that (b) and (c) are equivalent. We do this by showing that the negation of (b) and the negation of (c) are equivalent. Suppose $u \neq v$ are on a cycle of G. By Theorem 5.3, there are two paths from u to v. Conversely, suppose there are two paths from u to v. Call them $u = x_0, x_1, \ldots, x_k = v$ and $u = y_0, y_1, \ldots, y_m = v$. Let i be the smallest index such that $x_i \neq y_i$. We may assume that $i = 1$ for, if not, redefine $u = x_{i-1}$. On the new paths, let $x_a = y_b$ be the smallest $a > 0$ for which some x_j is on the y path. The walk

$$u = x_0,\ x_1, \ldots, x_a = y_b,\ y_{b-1}, \ldots, y_0 = u$$

has no repeated vertices except the first and last and so is a cycle. (A picture may help you visiualize what is going on. Draw the x path intersecting the y path several times.)

We now prove that (d) implies (b). Suppose that G has a cycle, $v_0, v_1, \ldots, v_k, v_0$. Remove the edge $\{v_0, v_k\}$. In any walk that uses that edge, replace it with the path $v_0, v_1, \ldots, v_k$ or its reverse, as appropriate. Thus the graph is still connected and so the edge $\{v_0, v_k\}$ contradicts (d).

5.4.2 (a) This is a restatement of the equivalence of (b) and (d) in the theorem.

(b) This was done in the last part of the proof of the theorem.

(c) Again, this was done in the last part of the proof of the theorem.

5.4.3 (a) By Exercise 5.1.1, we have $\sum_{v \in V} d(v) = 2|E|$. By 5.4(e), $|E| = |V| - 1$. Since

$$2|V| \;=\; \sum_{v \in V} 2, \quad \text{we have} \quad 2 \;=\; 2|V| - 2|E| \;=\; \sum_{v \in V} (2 - d(v)).$$

(b) We give three solutions. The first uses the previous result. The second uses the fact that each tree except the single vertex has at least two leaves. The third uses the fact that trees have no cycles.

Suppose that T is more than just a single vertex. Since T is connected, $d(v) \neq 0$ for all v. Let n_k be the number of vertices of T of degree k. By the previous result, $\sum_{k \geq 1} (2 - k) n_k = 2$. Rearranging gives $n_1 = 2 + \sum_{k \geq 2} (k - 2) n_k$. If $n_m \geq 1$, the sum is at least $m - 2$.

For the second solution, remove the vertex of degree m to obtain m separate trees. Each tree is either a single vertex, which is a leaf of the original tree, or has at least two leaves, one of which must be a leaf of the original tree.

For the third solution, let v be the vertex of degree m and let $\{v, x_i\}$ be the edges containing v. Each path starting v, x_i must eventually reach a leaf since there are no cycles. Call the leaf y_i. These leaves are distinct since, if $y_i = y_j$, the walk $v, x_i, \ldots, y_i = y_j, \ldots, x_j, v$ would lead to a cycle.

(c) Let the vertices be u and v_i for $1 \leq i \leq m$. Let the edges be $\{u, v_i\}$ for $1 \leq i \leq m$.

(d) Let $N = n_3 + n_4 + \cdots$, the number of vertices of degree 3 or greater. Note that $k - 2 \geq 1$ for $k \geq 3$. By our earlier formula, $n_1 \geq 2 + N$. If $n_2 = 0$, $N = |V| - n_1$ and so we have $n_1 \geq 2 + |V| - n_1$. Thus $n_1 \geq 1 + |V|/2$. Similarly, if $n_2 = 1$, $N = |V| - n_1 - 1$ and, with a bit of algebra, $n_1 \geq (1 + |V|)/2$.

(e) A careful analysis of the previous argument shows that the number of leaves will be closest to $|V|/2$ if we avoid vertices with high degrees. Thus we will try to make our vertices of degree three or less. We will construct some RP-trees, T_k with k leaves. Let T_1 the isolated vertex. For $k > 1$, let T_k have two children, one a single vertex and the other the root of T_{k-1}. Clearly T_k has one more leaf and one more nonleaf than T_{k-1}. Thus the difference between the number of leaves and nonleaves is the same for all T_k. For T_1 it is one.

5.4.4 (a) Suppose G is a graph with v vertices and v edges. By Theorem 5.4(b,e), the graph has a cycle. This proves the base case, $n = 0$. Suppose $n > 0$ and G is a graph with v vertices and $v + n$ edges. By the theorem again, we know that the graph has a cycle. By the proof of the theorem, we know that removing an edge from a cycle does not disconnect the graph. However, removing the edge destroys any cycles that contain it. Hence the new graph G' contains one less edge and at least one less cycle than G. By the induction hypothesis, G' has at least n cycles. Thus G has at least $n + 1$ cycles.

(b) Let G be a graph with components $G_1, \ldots, G_k$. With subscripts denoting components, G_i has v_i vertices, $e_i = v_i + n_i$ edges and at least $n_i + 1$ cycles. From the last two formulas, G_i has at least $1 + e_i - v_i$ cycles. Now sum over i.

(c) There are many possibilities. Here's one solution. The vertices are v and, for $0 \le i \le n$, x_i and y_i. The edges are $\{v, x_i\}$, $\{v, y_i\}$, and $\{x_i, y_i\}$. (This gives $n + 1$ triangles joined at v.) There are $1 + 2(n + 1)$ vertices, $3(n + 1)$ edges, and $n + 1$ cycles.

5.4.5. Since the tree has at least 3 vertices, it has at least $3 - 1 = 2$ edges. Let $e = \{u, v\}$ be an edge. Since there is another edge and a tree is connected, at least one of u and v must lie on another edge besides e. Suppose that u does. It is fairly easy to see that u is a cut vertex and that e is a cut edge.

5.4.6 (a) No such tree exists. A tree with six vertices must have five edges.

(b) No such tree exists. Such a tree must have at least one vertex of degree three or more and hence at least three vertices of degree one.

(c) There are many; for example, 8 vertices forming a cycle and two vertices of degree 0.

(d) No such graph exists. If it did, the components would be trees since there are no cycles. Consider the case of two trees. Suppose one tree has v vertices and the other $12 - v$. Since trees have one fewer edges than vertices, the total number of edges is

$$(v - 1) + (12 - v - 1) \;=\; 10$$

For more than two trees there would be even fewer edges.

(e) A tree with 6 vertices has 5 edges. Since the sum of the degrees of the vertices must be twice the number of edges, the sum of the degrees must be 10.

(f) Such a graph must have at least $1 + e - v = 1 + 6 - 4 = 3$ cycles.

(g) No such graph exists. If the graph has no cycles, then each component is a tree. In such a graph, the number of vertices is strictly greater than the number of edges.

5.4.7 (a) The idea is that for a rooted planar tree of height h, having at most 2 children for each non-leaf, the tree with the most leaves occurs when each non-leaf vertex has exactly 2 children. You should sketch some cases and make sure you understand this point. For this case $l = 2^h$ and so $\log_2(l) = h$. Any other rooted planar tree of height h, having most 2 children for each non-leaf, is a subtree (with the same root) of this maximal-leaf binary tree and thus has fewer leaves.

(b) The height h can be arbitrarily large.

(c) $h = l - 1$.

(d) $\lceil \log_2(l) \rceil$ is a lower bound for the height of *any* binary tree with l leaves. It is easy to see that you can construct a full binary tree with l leaves and height $\lceil \log_2(l) \rceil$.

(e) $\lceil \log_2(l) \rceil$ is the minimal height of a binary tree.

5.4.8. We'll give four proofs. The case $n = 1$ is trivial since the tree is just • in this case.

(i) By Exercise 5.4.3, $\sum_{v\in V}(2 - d(v)) = 2$. Since there is more than one leaf, the root r is not a leaf and so $d(r) = 2$. For a leaf u, $d(u)$. For any other vertex w, $d(w) = 3$. Let there be m nonleaf vertices. We have

$$2 = \sum_{v\in V}(2 - d(v)) = (2-2) + n(2-1) + m(2-3) = n - m.$$

Hence $m = n - 2$. The total number of vertices is $1 + n + m = 2n - 1$.

(ii) We give a proof by induction on the number of leaves. The case $n = 1$ was done. Otherwise, the binary tree consists of a root joined to two other binary trees. Let the left tree have k leaves. Then the right has $n - k$. By induction, the left tree has $2k - 1$ vertices and the right has $2(n - k) - 1$. Adding these together and adding 1 for the root gives a total of $2n - 1$ vertices.

(iii) Every non-leaf vertex has two children. Thus the total number of vertices equals 1 (for the root) plus twice the number of vertices which are not leaves. If there are v vertices, $v - n$ are not leaves and so $v = 1 + 2(v - n)$. Solving, we get $v = 2n - 1$.

(iv) The number of edges is twice the number of nonleaf vertices. If there are v vertices and e edges, then $e = 2(v - n)$. By Theorem 5.4(e), $e = v - 1$. Thus $v - 1 = 2(v - n)$. Solving, we get $v = 2n - 1$.

5.4.9 (a) A binary tree with 35 leaves and height 100 is possible.

(b) A full binary tree with 21 leaves can have height at most 20. So such a tree of height 21 is impossible.

(c) A binary tree of height 5 can have at most 32 leaves. So one with 33 leaves is impossible.

(d) A full binary tree with 65 leaves has minimal height $\lceil \log_2(65)\rceil = 7$. Thus a full binary tree with 65 leaves and height 6 is impossible.

5.4.10. The maximal number of vertices is $1 + k + k^2 + \cdots k^h = (k^{h+1} - 1)/(k - 1)$. The maximal number of leaves is k^h.

5.4.11 (a) Breadth-first: $MIAJKCEHLBFGD$,
Depth-first: $MICIEIHFHGHDHIMAMJMKLKBKM$,
Pre-order: $MICEHFGDAJKLB$,
Post-order: $CEFGDHIAJLBKM$.

(b) The tree is the same as in part (a), reflected about the vertical axis, with vertices A and J removed.

(c) It is not possible to reconstruct a rooted plane tree given just its pre-order vertex list. A counterexample can be found using just three vertices.

(d) It is possible to reconstruct a rooted plane tree given its pre-order and post-order vertex list. If the root is X and the first child of the root is Y, it is possible to reconstruct the pre-order and post-order vertex lists of the subtree rooted at Y from the pre-order and post-order vertex lists of the tree. In the same manner, you can reconstruct the pre-order and post-order vertex lists of the subtrees rooted at the other children of the root X. Now do the same trick on these subtrees. Try this approach on an example.

5.4.12. The statement of the exercise associates a Prüfer sequence with every tree. To prove that this is a bijection, we must show that it is an injection and a surjection. If we use the fact that there are n^{n-2} trees, we do not need to do both: A function $f : A \to B$ with $|A| = |B|$ is a surjection if and only if it is an injection. On the other hand, if we want to prove that there are n^{n-2} trees, we need to do both.

Note first that it does not matter what labels are used for the vertices as long as the labels are ordered so we can find the largest. Suppose the hint has been proved.

The proof of the bijection is by induction on the number of vertices. Given a Prüfer sequence, we can determine the first vertex removed. Thus we know which vertices are left. Furthermore, the Prüfer sequence for this $(n-1)$-vertex tree is the original sequence with the first entry removed. Using this you should be able to prove inductively that every Prüfer sequence gives a unique tree. Thus the function from trees to Prüfer sequences is a bijection.

We now prove the suggestion in the hint. Clearly the first vertex removed cannot be in the Prüfer sequence. Consider a vertex *not* in the Prüfer sequence. Since it does not appear in the sequence, no vertex attached to it was ever removed. Hence it must be a leaf. Since the largest leaf is removed, we are done.

Section 5.5

5.5.1. Let D be the domain suggested in the hint and define $f\colon D \to \mathcal{P}_2(V)$ by $f((x,y)) = \{x,y\}$. Let $G(D) = (V,\psi)$ where $\psi(e) = f(\varphi(e))$.

5.5.2. In each case, it is a matter of choosing subsets of $V \times V$ for the directed edges.

(a) There are $|V \times V|$ potential edges to choose from. Since there are two choices for each edge (either in the digraph or not), we get 2^{n^2} simple digraphs.

(b) With loops forbidden, our possible edges include all elements of $V \times V$ except those of the form (v,v) with $v \in V$. Thus there are $2^{n(n-1)}$ loopless simple digraphs. An alternative derivation is to note that a simple graph has $\binom{n}{2}$ edges and we have 4 possible choices in constructing a digraph: (i) omit the edge, (ii) include the edge directed one way, (iii) include the edge directed the other way, and (iv) include two egdes, one directed each way. This gives $4^{\binom{n}{2}} = 2^{n(n-1)}$. The latter approach is not useful in doing part (c).

(c) Given the set S of possible edges, we want to choose q of them. This can be done in $\binom{|S|}{q}$ ways. In the general case, the number is $\binom{n^2}{q}$ and in the loopless case it is $\binom{n(n-1)}{q}$

5.5.3. Let $V = \{u,v\}$ and $E = \{(u,v),(v,u)\}$.

5.5.4. For each $\{u,v\} \in \mathcal{P}_2(V)$ we have three choices: select the edge (u,v), select the edge (v,u) or have no edge between u and v. Let $N = |\mathcal{P}_2(V) = \binom{n}{2}$.

(a) There are 3^N oriented simple graphs.

(b) We can choose q elements of $\mathcal{P}_2(V)$ and then orient each of them in one of two ways. This gives us $\binom{N}{q}2^q$.

5.5.5. You can use the notation and proof of Example 5.5 provided you change all references to two element sets to references to ordered pairs. This means replacing $\{x,y\}$ with (x,y), $\{\nu(x),\nu(y)\}$ with $(\nu(x),\nu(y))$ and $\mathcal{P}_2(V_i)$ with $V_i \times V_i$.

5.5.6 (a) If $x,y \in V$ and $x \neq y$, there must be a directed pat from x to y in D since D is strongly connected. In $S(D)$, this becomes a walk from x to y. Hence $S(D)$ is connected.

(b) Here's the simplest solution: $V = \{x,y\}$ and $E = \{(x,y)\}$.

(c) Let $u_1 \in V_1$ and $u_2 \in V_2$. Since D is strongly connected, there is a directed path from u_1 to u_2. This path must somehow cross from V_1 to V_2 and so there is an edge from V_1 to V_2. Similarly, there's an edge from V_2 to V_1.

If you prefer, here's a more formal proof. Let v_1 be the last vertex on the path that is in V_1. Since $u_2 \notin V_1$, v_1 is not the end of the path. Let v_2 be the next vertex on the path after v_1.

By the definition of v_1, we have $v_2 \notin V_1$ and so $v_2 \in V_2$. Since $v_1, v_2)$ is an edge of D, there is an edge from V_1 to V_2. Interchanging the roles of V_1 and V_2 proves that there is also an edge from V_2 to V_1.

(d) Here are two different proofs.
First proof. Let u be a vertex and let V_1 contain all vertices that can be reached from u. Let V_2 be the remaining vertices. If $V_2 = \emptyset$, we are done. We now suppose $V_2 \neq \emptyset$ and obtain a contradiction. Since D is 2-way joined, there is an edge (w_1, w_2) with $w_1 \in V_1$ and $w_2 \in V_2$. By the definition of V_1, either $w_1 = u$ or there is a directed path from u to w_1. Since (w_1, w_2) is an edge, it follows that there is a directed path from u to w_2. This contradicts the definition of V_2 as those those vertices which cannot be reached from u. It follows that the assumption $V_2 \neq \emptyset$ is false.
Second proof. Suppose that D is not strongly connected. We will find a partition that is not 2-way joined. That will prove the result by proving the contrapositive. By assumption, there must be vertices v_1 and v_2 in D such that there is no directed path from v_1 to v_2. Let V_1 contain v_1 and all vertices that can be reached from v_1 via directed paths. Let V_2 be the remaining vertices. Since $v_2 \in V_2$, we have a partition of V. If (v, w) were an edge from V_1 to V_2, the path from v_1 to v followed by the edge (v, w) would give us a contradiction. Since no such edge (v, w) can exist, the partition is not 2-way joined.

5.5.7. "The statements are all equivalent" means that, given any two statements v and w, we have a proof that v implies w. Suppose D is strongly connected. Then there is a directed path $v = v_1, v_2, \ldots, v_k = w$. That means we have proved v_1 implies v_2, that v_2 implies v_3 and so on. Hence v_1 implies v_k.

5.5.8 (a) The value of $d_{\text{in}}(\{v\})$ is the number of edges that have their "heads" at v.

(b) Both sums equal the number of edges in D.

(c) Suppose $u \in U$. Every edge (v, u) contributes 1 to $d_{\text{in}}(\{v\})$ but it contributes 1 to $d_{\text{in}}(U)$ only when $v \notin U$. Hence $\sum_{u \in U} d_{\text{in}}(\{u\})$ exceeds $d_{\text{in}}(U)$ by the number of edges (v, u) with $v \in U$ and $u \in U$, which is what we were asked to prove.

(d) There is a result like (c) for d_{out}. Let $e(U)$ be the number of edges in D that have both their end points in U. We have

$$\begin{aligned} d_{\text{in}}(U) &= \sum_{u \in U} d_{\text{in}}(\{u\}) - e(U) \\ &= \sum_{u \in U} d_{\text{out}}(\{u\}) - e(U) \;=\; d_{\text{out}}(U). \end{aligned}$$

5.5.9. Let $e = (u_1, u_2)$. For $i = 2, 3, \ldots$, as long as $u_i \neq u_1$ choose an edge (u_i, u_{i+1}) that has not be used so far. It is not hard to see that $d_{\text{in}}(u_i) = d_{\text{out}}(u_i)$ implies this can be done. In this way we obtain a directed trail starting and ending at u_1. This may not be a cycle, but a cycle cotaining e can be extracted from it by deleting some edges.

5.5.10. Use the idea in the previous exercise plus induction on the number of edges to partition the edges of D into directed trails with the starting vertex of each trail equal to its final vertex. We will prove that if there is more than one trail, then two of the trails can be combined into a single trail. It follows that we may assume there is only one trail. To prove our combining claim, note that since $S(D)$ is connected, there must be two trails that have a vertex in common. It is not hard to see how to join them into one directed trail.

5.5.11 (a)

(b) See Exercise 6.3.14 (p. 166).

5.5.12 (a) There are 2^{n^2-n} reflexive binary relation on a set of n elements.

(b) There are $2^{(n^2-n)/2}$ reflexive and symmetric relations R on a set of n elements.

(c) There are $2^{(n^2-n)/2}$ unreflexive and symmetric relations R on a set of n elements.

(d) There are $2^{(n^2+n)/2}$ symmetric relations and $2^{(n^2-n)/2}$ reflexive and symmetric relations. Take the difference and you get $2^{(n^2-n)/2}(2^n-1)$.

5.5.13 (a) For all $x \in S$, $x|x$. For all $x, y \in S$, if $x|y$ and $x \neq y$, then y does not divide x. For all $x, y, z \in S$, $x|y$, $y|z$ implies that $x|z$.

(b) The covering relation

$$H = \{(2,4),(2,6),(2,10),(2,14),(3,6),(3,9),(3,15),(4,8),(4,12),(5,10),(5,15),(6,12),(7,14)\}.$$

5.5.14. The transitive closure of H is the divides relation on $S = \{2,3,4,5,6,7,8,9,10,11,12,13,14,15,16\}$.

5.5.15 (a) There are n^{n-2} trees. Since a tree with n vertices has $n-1$ edges, the answer is zero if $q \neq n-1$. If $q = n-1$, there are $\binom{\binom{n}{2}}{n-1}$ graphs. Thus the answer is $n^{n-2}\binom{\binom{n}{2}}{n-1}^{-1}$ when $q = n-1$.

(b) We have

$$\binom{\binom{n}{2}}{n-1} < \frac{\binom{n}{2}^{n-1}}{(n-1)!} = \frac{n^{n-1}(n-1)^{n-1}}{2^{n-1}\,(n-1)!} < \frac{n^{n-1}}{2^{n-1}/e^{n-1}} = \left(\frac{ne}{2}\right)^{n-1}.$$

Using this in the answer to (a) gives the result we want. It turns out that

$$n^{n-2}\binom{\binom{n}{2}}{n-1}^{-1} \sim \sqrt{\pi/2n}\,(2/e)^n,$$

which differs from our estimate by a constant times $n^{1/2}$.

Section 5.6

5.6.1. Let A and B be the partition of the vertices guaranteed by the definition of a bipartite graph. Let $k = |A|$, number the vertices in A with 1 to k and those is B with $k+1$ to n. Since no edges connect vertices in A to each other, $A(G)$ has a $k \times k$ block of zeroes in its upper left corner. Similarly B gives a block in the lower right corner.

5.6.2. Since there are no edges between connected components of G, we have lots of zeroes, more specifically, $A(G)$ is a matrix of zeroes except for blocks along the diagonal. The ith block is the $n_i \times n_i$ matrix $A(G_i)$.

5.6.3 (a) $a_{i,j}^{(k)}$ is the sum over all $t_1, \ldots, t_{k-1}$ of $a_{i,t_1}a_{t_1,t_2}\cdots a_{t_{k-1},j}$. Each of these products is 0 or 1, so the sum is nonzero if and only if some product is nonzero. This happens if and only if each factor in the product is nonzero. This happens if and only if the vertices $i, t_1, \ldots, t_{k-1}, j$ form a walk.

(b) We can construct a path from a walk by jumping over pieces that form cycles. Thus the shortest walk from i to j is a path. Here's a more formal argument. Suppose that $W = (i, t, \ldots, v, j)$ is the shortest walk from i to j. If it is not a path, then there must be repeated vertices in the list. Let u be such a vertex. Remove all vertices from the sequence after the first occurence of

u up to and including the last occurence of u. The result is a shorter walk, contradicting the minimality of W.

(c) The obvious idea is to repeat the previous statement with $i = j$: "The shortest walk from i to i is a cycle." This is not true. If $\{i, j\}$ is an edge, then i, j, i is the shortest walk from i to j but it is not a cycle. The result would be true if we were looking at oriented simple graphs because an edge can be traversed in only one direction. All we can claim is that any odd length walk from i to i contains a cycle.

We can modify the situation a bit by looking at an edge $\{i, j\}$ of the graph. Let H be the graph obtained by removing it; i.e., by setting $a_{i,j} = a_{j,i} = 0$. The shortest walk from j to i in H together with the edge $\{i, j\}$ is a cycle of G. This follows from the previous result and the definitions of path and cycle.

(d) Following the hint, $B^k = \sum_{t=0}^{k} \binom{k}{t} B^t$ by the binomial theorem. Since $\binom{k}{t} > 0$, $b_{i,j}^{(k)}$ is nonzero if and only if $a_{i,j}^{(t)} \neq 0$ for some t with $0 \le t \le k$. $t = 0$ gives the identity matrix, so $b_{i,i}^{(k)} \neq 0$ for all k. For $i \neq j$, $b_{i,j}^{(k)} \not 0$ if and only if there is a walk from i to j for some $t \le k$, and thus if and only if there is a path for some $t \le k$. Since paths of length t contain $t + 1$ distinct vertices, no path is longer than $n - 1$. Thus there is a path from i to $j \neq i$ if and only if $b_{i,j}^{(k)} \neq 0$ for all $k \ge n - 1$.

5.6.4. The arguments given for simple graphs carry over. Nothing can be said about cycles for simple directed graphs.

5.6.5. We claim that $A(D)$ is nilpotent if and only if there is no vertex i such that there is a walk from i to i (except the trivial walk consisting of just i).

First suppose that there is a nontrivial walk from i to i containing k edges. Let $C = A(D)^k$. It follows that all entries of C are nonnegative and $c_{i,i} \neq 0$. Thus $c_{i,i}^{(m)} \neq 0$ for all $m > 0$. Hence $A(D)$ is not nilpotent.

Conversely, suppose that $A(D)$ is not nilpotent. Let n be the number of vertices in D and suppose that i and j are such that $a_{i,j}^{(n)} \neq 0$, which we can do since $A(D)$ is not nilpotent. There must be a walk $i = v_0, v_1, v_2, \ldots, v_n = j$. Since this sequence contains $n + 1$ vertices, there must be a repeated vertex. Suppose that $k < l$ and $v_k = v_l$. The sequence $v_k, v_{k+1}, \ldots, v_l$ is a nontrivial walk from v_k to itself.

Section 6.1

6.1.1 (a) One description of a tree is: a connected graph such that removal of any edge disconnects the tree. Since an edge connects only two vertices, we will obtain only two components by removing it.

(b) Note that T with e removed and f added is a spanning tree. Since T has minimum weight, the result follows.

(c) The graph must have a cycle containing e. Since one end of e is in T_1 and the other in T_2, the cycle must contain another connector besides e.

(d) Since T^* with e removed and f added is a spanning tree, the algorithm would have removed f instead of e if $\lambda(f) > \lambda(e)$.

(e) By (b) and (d), $\lambda(f) = \lambda(e)$. Since adding f connects T_1 and T_2, the result is a spanning tree.

(f) Suppose T^* is not a minimum weight spanning tree. Let T be a minimum weight spanning tree so that the event in (a) occurs as late as possible. It was proven in (e) that we can replace T with another minimum weight spanning tree such that the disagreement between T and T^*, if any, occurs later in the algorithm. This contradicts the definition of T.

6.1.2. Note that B_1 and B_2 cannot have any edges in common since they are different equivalence classes of edges. Suppose that u and v are vertices in $B_1 \cap B_2$. We will derive a contradiction by finding a cycle containing edges of B_1 and edges of B_2. There exist vertices $w_{i,1}$ such that $e_i = \{u, w_{i,1}\}$ is an edge of B_i. There also exist edges f_i in B_i that have v as an end vertex. Since $e_i \sim f_i$, there is a cycle in B_i containing e_i and f_i. Let the vertices on the cycle in B_i be $u, w_{i,1}, w_{i,2}, \ldots, v, \ldots$. Let x be the first vertex in the B_1 cycle other that u which lies in the B_2 cycle. (There must be one since both cycles contain v.) If $x = w_{1,1} = w_{2,1}$, we are done since then $e_1 = e_2$ is in both B_1 and B_2, a contradiction. Then

$$u, w_{1,1}, \ldots, x, \ldots, w_{2,j}, w_{2,j-1}, \ldots, w_{2,1}, v$$

is a cycle containing e_1 and e_2. Thus $e_1 \sim e_2$, a contradiction.

6.1.3 (b) Let Q_1 and Q_2 be two bicomponents of G, let v_1 be a vertex of Q_1, and let v_2 be a vertex of Q_2. Since G is connected, there is a path in G from v_1 to v_2, say $x_1, \ldots, x_p$. You should convince yourself that the following pseudocode constructs a walk $w_1, w_2, \ldots$ in $\mathcal{B}(G)$ from Q_1 to Q_2.

```
Set w_1 = Q_1, j=2, and k = 0.
While there is an x_i ∈ P(G) with i > k.
   Let i > k be the least i for which x_i ∈ P(G).
   If i = p
      Set Q = Q_2.
   Else
      Let Q be the bicomponent containing {x_i, x_{i+1}}.
   End if
   Set w_j = x_i, w_{j+1} = Q, k = i, and j = j + 2.
End while
```

(c) Suppose there is a cycle in $\mathcal{B}(G)$, say $v_1, Q_1, \ldots, v_k, Q_k, v_1$, where the Q_i are distinct bicomponents and the v_i are distinct vertices. Set $v_{k+1} = v_1$. By the definitions, there is a path in Q_i from v_i to v_{i+1}. Replace each Q_i in the previous cycle with these paths after removing the endpoints v_i and v_{i+1} from the paths. The result is a cycle in G. Since this is a cycle, all vertices on it lie in the same bicomponent, which is a contradiction since the original cycle contained more than one Q_i.

(d) Let v be an articulation point of the simple graph G. By definition, there are vertices x and y such that every path from x to y contains v. From this one can prove that there are edges $e = \{v, x'\}$ and $f = \{v, y'\}$ such that every path from x' to y' contains v. It follows that e and f are in different bicomponents. Thus v lies in more than one bicomponent.

Suppose that v lies in two bicomponents. There are edges $e = \{v, w\}$ and $f = \{v, z\}$ such that $e \not\sim f$. It follows that every path from w to z contains v and so v is an articulation point.

6.1.4. When we removed e from the minimum weight spanning tree and added f, the result was still a minimum weight spanning tree. It follows that $\lambda(e) = \lambda(f)$, which contradicts $e \neq f$.

6.1.5 (a) Since there are no cycles, each component must be a tree. If a component has n_i vertices, then it has $n_i - 1$ edges since it is a tree. Since $\sum n_i$ over all components is n and $\sum(n_i - 1)$ over all components is k, $n - k$ is the number of components.

(b) By the previous part, H_{k+1} has one less component than G_k does. Thus at least one component C of H_{k+1} has vertices from two or more components of G_k. By the connectivity of C, there must be an edge e of C that joins vertices from different components of G_k. If this edge is added to G_k, no cycles arise.

(c) By the definition of the algorithm, it is clear that $\lambda(g_1) \le \lambda(e_1)$. Suppose that $\lambda(g_i) \le \lambda(e_i)$ for $1 \le i \le k$. By the previous part, there is some e_j with $1 \le j \le k+1$ such that G_k together with e_j has no cycles. By the definition of the algorithm, it follows that $\lambda(g_{k+1}) \le \lambda(e_j)$. Since $\lambda(e_j) \le \lambda(e_{k+1})$ by the definition of the e_i's, we are done.

6.1.6. In the notation of the previous exercise, the edges e_i belong to a minimum weight spanning tree if and only if $\lambda(g_i) = \lambda(e_i)$ for $1 \le i < n$. Since λ is an injection, it follows that $g_i = e_i$ for $1 \le i < n$.

6.1.7 (a) Hint: For (1) there are four spanning trees. For (2) there are 8 spanning trees. For (3) there are 16 spanning trees.

(b) Hint: For (1) there is one. For (2) there are two. For (3) there are two.

(c) Hint: For (1) there are two. For (2) there are four. For (3) there are 6.

(d) Hint: For (1) there are two. For (2) there are three. For (3) there are 6.

6.1.8 (a) Hint: For *(1)* there are three minimal spanning trees. For *(2)* there are 2 spanning trees. For *(3)* there is 1 minimal spanning tree.

(b) Hint: For *(1)* there is one. For *(2)* there are two. For *(3)* there is one.

(c) Hint: *(1)* there is one. For *(2)* there is one. For *(3)* there are four.

(d) Hint: For *(1)* there is one. For *(2)* there is one. For *(3)* there are four.

6.1.9 (a) Hint: There are 21 vertices, so the minimal spanning tree has 20 edges. Its weight is 30.

(b) Hint: Its weight is 30..

(c) Hint: Its weight is 30.

(d) Hint: Note that K is a the only vertex in common to the two bicomponents of this graph. Whenever this happens (two bicomponents, common vertex), the depth-first spanning tree rooted at that common vertex has exactly two "principal subtrees" at the root. In other words, the root of the depth-first spanning tree has degree two. Finding depth first spanning trees of minimal weight is, in general, difficult. You might try it on this example.

Section 6.2

6.2.1. This is just a matter of a little algebra.

6.2.2. The answer is $x(x-1)^{n-1}$ for a tree with n vertices. We'll give a three methods.

First method. Imagine the tree being rooted and color it working from the root. The root can be colored in x ways. When we reach another vertex, its parent is the only vertex adjacent to it which has been colored, so it can be colored in $x-1$ ways.

Second method. We'll use induction on the number of vertices.. One vertex is trivial. For more than one vertex, the tree T must have a leaf v. By induction, $T-v$ can be colored in $x(x-1)^{n-2}$ ways. By definition, a leaf is a vertex which is joined to the rest of the tree by just one edge thus v can be colored in $x-1$ ways.

Third method. We'll use induction on the number of vertices.. One and two vertices are trivial. If there are more than two vertices in the tree T, it has a vertex v of degree greater than one. We can split T into two trees H and K which share only the vertex v and which each have less leaves than T. By Exercise 6.2.3 below, the result follows.

6.2.3 (a) To color G, first color the vertices of H AND then color the vertices of K. By the Rule of Product, $P_G(x) = P_H(x)P_K(x)$.

(b) Let v be the common vertex. There is an obvious bijection between pairs of colorings (λ_H, λ_K) of H and K with $\lambda_H(v) = \lambda_K(v)$ and colorings of G. We claim the number of such pairs is $P_H(x)\big(P_K(x)/x\big)$. To see this, note that, in the colorings of K counted by $P_K(x)$, each of the x ways to color v occurs equally often and so $1/x$ of the colorings will have $\lambda_K(v)$ equal to the color given by $\lambda_H(v)$.

(c) The answer is $P_H(x)P_K(x)(x-1)/x$. We can prove this directly, but we can also use (b) and (6.4) as follows. Let $e = \{v, w\}$. By the construction of G, $P_{G-e}(x) = P_H(x)P_K(x)$. By (b), $P_{G_e}(x) = P_H(x)P_K(x)/x$. Now apply (6.4).

6.2.4. By Exercise 6.2.3, $P_G(x) = P_{Z_{k+1}}(x)P_{Z_{n-k+1}}(x)/x(x-1)$.

6.2.5. Let the solution be $P_n(x)$. Clearly $P_1(x) = x(x-1)$, so we may suppose that $n \ge 2$. Apply deletion and contraction to the edge $\{(1,1),(1,2)\}$. Deletion gives a ladder with two ends sticking out and so its chromatic polynomial is $(x-1)^2P_{n-1}(x)$. Contraction gives a ladder with the contracted vertex joined to two adjacent vertices. Once the ladder is colored, there are $x-2$ ways to color the contracted vertex. Thus we have

$$P_n(x) \;=\; (x-1)^2P_{n-1}(x) - (x-2)P_{n-1}(x) \;=\; (x^2-3x+3)P_{n-1}(x).$$

The value for $P_n(x)$ now follows easily.

6.2.6. Use deletion and contraction on the edge $\{(1,2),(2,2)\}$ and then on the edge $\{(2,2),(3,2)\}$. Two contractions give two easy graphs that have a common vertex. A contraction and a deletion in either order gives Z_6 with two vertices joined. By coloring Z_6 and then those two, you get the chromatic polynomial for Z_6 times $(x-2)^2$. After two deletions, use the edge $\{(2,1),(2,2)\}$. Deletion gives Z_9 with a vertex joined to it by a single edge. Contraction gives two copies of Z_5 sharing an edge. All the graphs we have obtained have chromatic polynomials that are easy to compute by previous results.

6.2.7. The answer is

$$x^8 - 12x^7 + 66x^6 - 214x^5 + 441x^4 - 572x^3 + 423x^2 - 133x.$$

There seems to be no really easy way to derive this. Here's one approach which makes use of Exercise 6.2.3 and $P_{Z_n}(x)$ for $n = 3,4,5$. Label the vertices reading around one face with a,b,c,d and around the opposite face with A,B,C,D so that $\{a,A\}$ is an edge, etc. If the edge $\{a,A\}$ is contracted, call the new vertex α. Introduce β, γ and δ similarly.

Let $e_1 = \{a,A\}$ and $e_2 = \{b,B\}$. Note that $G - e_1 - e_2$ consists of three squares joined by common edges and that $H = G_{e_1} - e_2$ is equivalent to $(G-e_1)_{e_2}$. We do H in the next paragraph. In $K = G_{e_1e_2}$, let $f = \{\alpha, \beta\}$. $K - f$ is two triangles and a square joined by common edges and K_f is a square an a vertex v joined to the vertices of the square. By first coloring v and then the square, we see that $P_{K_f}(x) = xP_{Z_4}(x-1)$.

Let $f_1 = \{c, C\}$, $f_2 = \{d, D\}$ and $f_3 = \{\beta, \gamma\}$. Then

- $H - f_1 - f_2$ is two Z_5's sharing β;
- $(H - f_1)_{f_2}$ is easy to do if you consider two cases depending on whether β and δ have the same or different colors, giving $x(x-1)(x-2)^4 + x(x-1)^4$;
- $H_{f_1} - f_3$ is a Z_5 and a triangle with a common edge and
- $H_{f_1f_3}$ are three triangles joined by common edges.

6.2.8. A term of the sum on the right hand side of (6.5) counts the number of functions from $\underline{n}$ to $\underline{x}$ for which $|\text{Image}(f)| = k$.

6.2.9. This can be done by induction on the number of edges. The starting situation involves some number n of vertices with no edges. Since the chromatic polynomial is x^n, the result is proved for the starting condition.

Now for the induction. Deletion does not change the number of vertices, but reduces the number of edges. By induction, it gives a polynomial for which the coefficient of x^k is a nonnegative multiple of $(-1)^{n-k}$. Contraction decreases both the number of vertices and the number of edges by 1 and so gives a polynomial for which the coefficient of x^k is a nonnegative multiple of $(-1)^{n-1-k}$. Subtracting the two polynomials gives one where the coefficient of x^k is an nonnegative multiple of $(-1)^{n-k}$.

Section 6.3

6.3.1. Every face must contain at least four edges and each side of an edge contributes to a face. Thus $4f \ge$ (edge sides) $= 2e$. From Euler's relation,

$$2 \;=\; v - e + f \;\ge\; v - e + e/2 \;=\; (2v - e)/2$$

and so $e \ge 2v - 4$.

6.3.2. One can see that it contains no cycle of length 3. (Either study it or note that it is bipartite.) By the previous exercise, we must have $e \le 2v - 4$ if it is planar; however, $v = 6$ and $e = 9$.

6.3.3 (a) We have $2e = fd_f$ and $2e = vd_v$. Use this to eliminate v and f in Euler's relation.

(b) They are cycles.

(c) If $d_f \ge 4$ and $d_v \ge 4$, we would have $0 < 2/d_f + 2/d_v - 1 \le 0$, a contradiction. Thus at least one of d_v and d_f is 3. Since $d_v \ge 3$, we have $2/d_v \le 2/3$. Thus

$$0 \;<\; \frac{2}{d_f} + \frac{2}{d_v} - 1 \;\le\; \frac{2}{d_f} - \frac{1}{3}$$

and so $d_f \;<\; 2/(1/3) \;=\; 6$. Since d_f is an integer, $d_f \le 5$. Since $d_f \ge 3$ for a simple graph, interchanging f and v in the above gives us $d_v \le 5$.

(d) Altogether there are 5 possibilities for the pair (d_v, d_f) by the previous part of the exercise. Given d_f and d_v, we can solve (6.9) for e. Then $vd_v = 2e$ and $fd_f = 2e$ give v and f. The five graphs turn out to be the Platonic solids with the interiors removed. (They are the tetrahedron, cube, octahedron, dodecahedron and icosohedron.)

6.3.4. This pattern actually appears on a soccer ball, so one could simply get a soccer ball and count. We'll use Euler's relation.

Since each carbon atom is joined to 3 others it lies on 3 edges, there are $3v = 180$ ends of edges. Since each edge has 2 ends, $2e = 180$ and so $e = 90$. Suppose there are f_5 pentagons and f_6 hexagons. Since each edge appears on two faces, $5f_5 + 6f_6 = 2e = 180$. By Euler's relation, $f_5 + f_6 = e - v + 2 = 32$. Solving the pair of equations

$$5f_5 + 6f_6 = 180 \quad \text{and} \quad f_5 + f_6 = 32,$$

we obtain 12 pentagons and 20 hexagons.

6.3.5. The value of c is zero. Suppose when we cut as directed we cut through k edges. Each of these edges now becomes two, giving us k new edges. The same happens with the k faces. On each of the circles that we fill in with, we also get k edges and k vertices. The two circles give us 2 new faces. In summary, if we originally had $|V|$ vertices, $|E|$ edges and f faces on the torus, we now have a graph embedded on the sphere with $V| + 2k$ vertices, $|E| + k + 2k$ edges, and $f + k + 2$ faces. From Euler's relation on the sphere,

$$2 \;=\; (|V| + 2k) - (|E| + 3k) + (f + k + 2) \;=\; |V| - |E| + f.$$

Thus $|V| - |E| + f \;=\; 0$.

There's a subtle issue here: We described the cut as if each edge and face it encountered was different. This may not be the case, an edge (and face) can twist around the torus so that the cut meets it more than once; however, the counts are still correct. One way to see this is to imagine what happens if we cut around the face and stretch it flat. Stretching will distort our "bracelet cut" into some sort of curve that may cut through the face several times. Every time it passes through the face it creates another face, two edges and two vertices.

6.3.6 (b) *abcdace*.

(c) *abcdadc*.

6.3.7. One method is to list all the simple planar graphs with $V = \underline{5}$ and find the least colorings for them. We use a theoretical argument instead.

The lex least proper coloring of $\underline{k} \subseteq V$ uses at most the first k colors. If it uses all k colors, then vertex k must be connected to each of the other vertices and the first $k - 1$ vertices must use all of the first $k - 1$ colors.

Let's apply these observations with $k = 5$, 4, 3 and 2 to a graph whose lex least coloring takes 5 colors. With $k = 5$, we see that vertex 5 is connected to each of the first 4 vertices and they use 4 different colors. Now, with $k = 4$, we see that vertex 4 is connected to each of the first 3 vertices and they use 3 different colors. Doing the same thing with $k = 3$ and $k = 2$, we finally see that every vertex is connected to every other; i.e., the graph is K_5, which is not planar.

6.3.8. A solution is

$$\begin{aligned} E \;=\; \{&\{1,2\},\{1,3\},\{1,6\},\{2,3\},\{2,4\},\{2,5\},\\ &\{2,6\},\{3,4\},\{3,5\},\{3,6\},\{4,5\},\{5,6\}\}. \end{aligned}$$

6.3.9. The argument for degree 4 is correct. For degree 5, we can assume, perhaps after rotating and or flipping the graph, that $y_1, \ldots, y_5$ are assigned colors c_1, c_2, c_3, c_4 and c_2, respectively. Suppose we look at y_1 and y_3 as in the text. The argument given there is okay if we get y_1 and y_3 in separate components. If they are in the same component, we end up switching colors c_2 and c_4 in the component of the subgraph colored by c_2 and c_4 that contains v_4. The colors of $y_1, \ldots, y_4$ are now c_1, c_2, c_3 and c_2. If y_5 was not in the same component with y_4, it is colored c_2 and we are done. Unfortunately, if y_4 and y_5 are in the same component, *its color is switched to* c_4. You should convince yourself that there is no way to arrange things to avoid this possibility.

6.3.10. The figure on the torus is essentially unique, but if we try to draw it in the plane, there are many possibilities. Figure S.6.1 shows one possible picture.

6.3.14 (a) Direct each edge $\{u,v\}$ with $\lambda(u) < \lambda(v)$ as follows: If $\{u,v\} = \{s,t\}$, the edge is (v,u); otherwise, it is (u,v). We must show that, for all vertices x, y, there is a directed path from x to y. It suffices to prove that there is a directed walk from x to y. To do this, it suffices to show that for any vertex u there is a directed walk from s to t that contains u, for then we can walk from u to t, use the edge (t,s) and, finally, walk from s to v.

We now construct a directed walk from s to t through u. If $u = s$ or $u = t$, it suffices to replace u by any other vertex of G. (Such a vertex exists because G has at least two edges and so must have more than two vertices.) To get a walk from s to u, induct on $k = \lambda(u) - 1$ and to get one from u to t, induct on $k = n - \lambda(u)$. Definition 6.4(c) can be used to start the induction at $k = 1$ and to carry out the induction step.

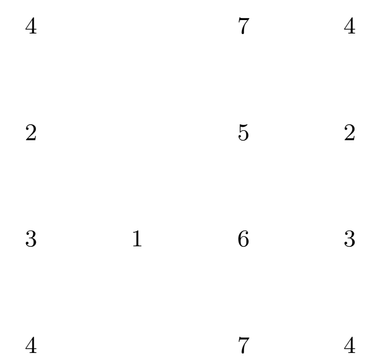


Figure S.6.1 A seven-color map on the torus for Exercise 6.3.10. Cut the square along the dashed lines and tape opposite sides together to form a torus. (To do this, the figure must be on stretchable material, like a rubber sheet.) Partial extra copies of the square are shown above and to the right of the square so you can more easily see where graph edges go.

(b) Suppose that $\{u, v\}$ is a bicomponent of G. Then no cycle contains $\{u, v\}$ and so removing the edge $\{u, v\}$ disconnects G; that is, $\{u, v\}$ is an isthmus of G.

(c) Let u and v be vertices of the graph G. Since G is connected, there is a path $u = u_1, \ldots, u_k = v$ in G. We will induct on k. If $k = 2$, then $\{u, v\}$ is an edge and belongs to some bicomponent of G and we are done by (a). Otherwise, let i be as large as possible so that u_i is in the same bicomponent as u. Then $i \geq 2$, there is a directed path from u to u_i by (a) and there is a directed path from u_i to u_k by the induction hypothesis. This gives us a directed walk from u to v.

6.3.15. We know from the text that a biconnected graph has an st-labeling. If $|V| = 2$, the result is trivial. Suppose that we have an st-labeling and that $\{x, y\}$ is an edge different from $\{s, t\}$. We may assume that $\lambda(x) < \lambda(y)$. By (iii) in the definition of st-labeling, we can find a sequence $y = w_1, w_2, \ldots = t$ such that $\lambda(w_i)$ is strictly increasing and such that $\{w_i, w_{i+1}\} \in E$. Similarly, we can find $x = u_1, u_2, \ldots = s$. These two paths with $\{s, t\}$ and $\{x, y\}$ form a cycle of G and so $\{x, y\}$ and $\{s, t\}$ are in the same bicomponent.

Section 6.4

6.4.1 (a) The value of a maximum flow is 45. Every maximum flow f will have $f(q, f) = 10$. Some other values of f are also determined uniquely, but many are not; for example, the flow into r can have any value from 15 to 20. Of course, the flows on the minimum cut set are unique. There are four minimum cut sets. The one found using $\mathcal{A}(f)$ is

$$\{\{r, h\}, \{f, a\}, \{k, e\}, \{y, u\}, \{z, u\}\}.$$

The others are obtained

(i) by deleting $\{r, h\}$ and adding $\{h, a\}$ and $\{h, c\}$,

(ii) by deleting $\{y, u\}$ and $\{z, u\}$ and adding $\{u, n\}$, or

(iii) by doing both (i) and (ii).

(b) See the previous solution.

(c) The value of a maximum flow is 25. Every maximum flow f will have $f(v,q)=10$. Some other values of f are also determined uniquely, but many are not. There is just one minimum cut set:

$$\{\{c,d\},\{k,e\},\{r,x\},\{w,x\}\}.$$

(d) See the previous solution. Since we do not have tools for finding all minimum cut sets, you may not have been able to prove that the minimum cut set was unique.

6.4.2. Let $(a,b) \in \text{FROM}(A,B)$ with $f(a,b) < c(a,b)$. By the definition of A, there is an augmentable path from some $D \in \mathcal{D}_{\text{in}}$ to a. Let δ be its increment. Since $b \notin A$, b is not on this path. By appending b to this path, we obtain an augmentable path with increment $\min(\delta, c(a,b)-f(a,b)) > 0$, implying that $b \in A$, a contradiction.

Now suppose that $(b,a) \in \text{FROM}(B,A)$ and $f(b,a) > 0$. As before, there is an augmentable path with end a and increment δ to which we can append b to obtain an augmentable path with increment $\min(\delta, f(b,a))$.

6.4.3. Since no complete augmentable path exists, $\mathcal{D}_{\text{in}} \subseteq A \subseteq V-\mathcal{D}_{\text{out}}$. Since $b(v)=0$ for $v \notin \mathcal{D}$, it follows that $\sum_{v\in A} b(v) = \sum_{v\in\mathcal{D}_{\text{in}}} b(v)$, which is the definition of the value of a flow. Recall that $b(v)$ is the sum of all flows out of v minus the sum of all flows into v. It follows that for $e=(x,y)\in E$, $b(x)$ has a contribution of $f(x,y)$ and $b(y)$ has a contribution of $-f(e)$. We distinguish four cases according as x and y are in A or B and ask what $f(e)$ contributes to $\sum_{v\in A} b(v)$.

(i) $x\in B$, $y \in B$: Then $f(e)$ contributes nothing to the sum.

(ii) $x \in A$, $y\in A$: Then $f(e)$ contributes both $f(e)$ and $-f(e)$, which gives a net contribution of zero.

(iii) $x\in A$, $y \in B$; i.e., $(e)\in \text{FROM}(A,B)$: Then $f(e)$ contributes $f(e)$ to the sum.

(iv) $x \in B$, $y\in A$; i.e., $(e) \in \text{FROM}(B,A)$: Then $f(e)$ contributes $-f(e)$ to the sum.

6.4.4. A maximum flow will have 5 liters/sec flowing into D_3 and 2 liters/sec flowing into D_4 from P_2. When the pumps worked better, we could put a greater demand on P_1 and so get 8 liters/sec flowing into D_3.

6.4.5 (a) Without examining the network in detail, we would need to let c'_1 and c'_2 (resp. c'_3 and c'_4) be the sum of the capacities of edges leaving (resp. entering) the corresponding P'_i. That way we can guarantee the capability of supplying (resp. removing) as much fluid as the pump could possibly send out to (resp. get in from) other other sources. If we know all the maximum flows for the original network, we may be able to improve on this: We need to set c'_i to the largest net flow out of (resp. into) D_i for all maximum flows in the original network. This leads to no improvement in this case.

(b) Yes. Let f' be a flow in the new network shown for the exercise. With the c'_i edges removed and the P'_i pumps converted back to depots. If we eliminate these edges from f' we obtain a flow f in the network of Figure 6.6. We'll have value(f) = value(f') because the sum of the net flows out of D_1 and D_2 for f equals the net flow out of D_0 for f' because $b(P'_1)=b(P'_2)=0$ for f'.

6.4.6. Let (A,B) be a cut partition and consider any directed path P from a source to a sink. The path starts in A and end in B. If x is the last vertex of P that is in A and y is the next vertex of P, then $(x,y)\in\text{FROM}(A,B)$. Since any path P from a source to a sink has an edge in FROM(A,B), this is a cut set.

Conversely, suppose that F is a cut set. Let A consist of $\mathcal{D}_{\text{in}}$ and all vertices that can be reached from $\mathcal{D}_{\text{in}}$ along a directed path containing no edges of F. Since F is a cut set, A contains no sinks and so (A,B) with $B=V-A$ is a cut partition. Suppose $e=(x,y)\in\text{FROM}(A,B)$. Since

$x \in A$, there is a path P from a source to x that contains no edge of F. Append e to this path. Since $y \notin A$, the new path contains an edge in F. Thus $e \in F$ and so $\text{FROM}(A, B) \subseteq F$. (Note that we generally do not have equality because F may be "too large;" e.g., we could have $F = E$.)

6.4.7. Let f and g be two maximum flows and let $A = \mathcal{A}(f)$. By the proof of the Augmentable Path Theorem, we see that $\text{value}(g) = \text{value}(f)$ if and only if $g(e) = c(e)$ for all $e \in \text{FROM}(A, B)$ and $g(e) = 0$ for all $e \in \text{FROM}(B, A)$. It is tempting to conclude that therefore $A = \mathcal{A}(g)$, but this does not follow immediately.

Here is a correct proof. As above, let $A = \mathcal{A}(f)$. If $A \neq \mathcal{A}(g)$, we can assume that there is some $v \in \mathcal{A}(g)$ with $v \notin A$. (If not, interchange the names of f and g.) Let $u_1, u_2, \ldots$ be an augmentable path for g that ends at v. Let δ be its increment. Since $v \notin A$, and $u_1 \in \mathcal{D}_{\text{in}} \subseteq A$, there is an i with $u_i \in A$ and $u_{i+1} \in B$. If $e = (u_i, u_{i+1})$ is the directed edge of G, then $g(e) \leq c(e) - \delta$ and $e \in \text{FROM}(A, B)$. If $e = (u_{i+1}, u_i)$ is the directed edge of G, then $g(e) \geq \delta$ and $e \in \text{FROM}(B, A)$. In either case, the idea in the previous paragraph proves that $\text{value}(g) < \text{value}(f)$, contradicting the assumption that g is a maximum flow.

6.4.8 (a) Let $a_1, \ldots, a_n$ be a system of distinct representatives. Since they are distinct, they must be a permutation of $\underline{n}$. Since $a_i \in A_i$, it is different from all other entries in the ith column of L.

(b) We have $s \in A_i$ if and only if s does not appear in the ith column of L. Since s appears once in each row, it appears exactly r times in L. Since each appearance is in a different column, it appears in r columns and so does *not* appear in exactly r of the A_i's.

(c) (Since the ith column contains r distinct integers $|A_i| = n-r$.) We use the Philip Hall Theorem. Suppose that $I \subseteq \underline{n}$. Form a list (ordered or not) by including the elements of each A_i for $i \in I$. Since $|A_i| = n-r$, the list has $(n-r)|I|$ elements, not all of By the previous part, each element appears on the list exactly $n-r$ times. Thus the list contains $(n-r)|I|/(n-r) = |I|$ distinct elements. (When $n = r$, this breaks down because we have $0/0$.)

(d) Suppose L is an $r \times n$ Latin rectangle. It can trivially be completed when $n - r = 0$, since it is already a square. Suppose that $r < n$. By the previous part of the exercise, we can add a row to L to obtain an $(r+1) \times n$ Latin rectangle L'. Since $n - (r+1) < n - r$, L' can be completed by induction.

6.4.9 (c) By induction, there is an SDR for the A_i, $i \in I$. If the claimed inequality is true, then there is also an SDR for the B_i, $i \in \underline{n} - X$. Taken together, these give us our representatives. It remains to prove the inequality. We have

$$\bigcup_{i \in I \cup R} A_i \;=\; \left(\bigcup_{i \in R} B_i\right) \cup X,$$

where the last union is disjoint. Thus

$$\left|\bigcup_{i \in R} B_i\right| \;=\; \left|\bigcup_{i \in R \cup I} A_i\right| - |X| \;\geq\; |R \cup I| - |I| \;=\; |R|.$$

6.4.10. Following the hint, we make a network with one source u and one sink v. Let each edge have capacity 1. Given a set of directed edge disjoint paths, define a flow f by setting $f(e) = 1$ if e belongs to any of the paths in the set and $f(e) = 0$ otherwise. You should be able to see that this is indeed a flow and that $\text{value}(f)$ is the number of paths. Thus the maximum number of edge disjoint paths does not exceed the value of the network's maximum flow. We will show how to associate $\text{value}(g)$ edge disjoint paths an integer valued flow g. The Integer Flow and Max-Flow Min-Cut Theorems will complete the proof.

We'll show how to build up a set of edge disjoint paths from g. Choose an edge $e_1 = (u, u_1)$ with $u_1 \in \mathcal{D}_{\text{in}}$ and $g(e_1) = 1$. Since g is a flow, either $u_1 \in \mathcal{D}$ or there is an edge $e_2 = (u_1, u_2)$ with $g(e_2) = 1$. Proceeding in this manner, we obtain a directed path $e_1, \ldots, e_k = (u_{k-1}, u_k)$ with $u_k \in \mathcal{D}$. Set $g'(e_i) = 0$ for $1 \le i \le k$ and $g'(e) = g(e)$ otherwise to obtain a new flow. If $u_k = u$, $\text{value}(g') = \text{value}(g)$. If $u_k = v$, $\text{value}(g') = \text{value}(g) - 1$ and we add the path to our set of paths. Iterating this process, we eventually reach a flow of with value zero and a set of value(g) edge disjoint paths from u to v.

6.4.11. The result in the previous exercise is valid when all edges are taken to be undirected. To see this, construct a directed graph by replacing each edge $\{x, y\}$ of G with the two edges (x, y) and (y, x). The first part of the previous proof goes through. If a directed path $e_1, e_2, \ldots$ is constructed from a flow, replace each edge (x, y) in the directed path with $\{x, y\}$. This gives what we will call a pseudo-path. The same edge may appear twice in the pseudo-path because there may be two directed edges $e_i = (x, y)$ and $e_j = (y, x)$ which give the same undirected edge. We may assume that $i < j$. Replace the pseudo-path with the pseudo-path obtained from $e_1, \ldots, e_{i-1}, e_{j+1}, \ldots$. Iterating this process eventually leads to a path from u to v. (You may want to fill in some details about that.)

6.4.12. The problem pretty well states what is to be proved. For the undirected case, first convert it to a directed graph as done in the previous exercise. For the directed case, split each vertex x as done in Exercise 6.4.4 to obtain a pair of vertices and an edge (x', x'') connecting them. Let the source be u'' and the sink v'. When a directed path is obtained, reverse the steps. Note that each vertex can appear in at most one directed path because $c(x', x'') = 1$ in the altered graph.

Section 6.5

6.5.1 (a) The probability that a vertex v has degree d is $\binom{n-1}{d} p^d (1-p)^{n-1-d}$ since we must choose d of the remaining $n-1$ vertices to connect to v, then multiply by the probability of an edge being present (p) or absent ($1-p$). Probabilities multiply since edges are independent in $\mathcal{G}_p(n)$. Using linearity of expectation and summing over all n vertices, we get $n\binom{n-1}{d} p^d (1-p)^{n-1-d}$.

(b) If C is a potential 4-cycle of 4 vertices, let $X_C = 1$ if the cycle is present and $X_C = 0$ if it is not. Then $\mathbf{E}(X_C) = p^4$. We must multiply this by the number of choices for C; that is, the number of potential 4-cycles. This number is $\binom{n}{4} \times 3 = \frac{n(n-1)(n-2)(n-3)}{4\times 2}$, which can be derived in at least two ways:

- Note that there are 3 ways to make a 4-cycle out of a set of 4 vertices.
- Choose an ordered list of 4 vertices that represent walking around a cycle. There are 4 vertices that could have been chosen as the starting vertex and 2 ways we could have gone around the cycle.

(c) This is the same as the previous situation, except that now we must make sure the two edges that cut across the 4-cycle are not present. Hence the answer is $3\binom{n}{4} p^4 (1-p)^2$.

6.5.2 (a) For each injection $\varphi : V_H \to \underline{n}$, let $X_\varphi = 1$ if the injection is an embedding of H into the random graph and let $X_\varphi = 0$ otherwise. Then $\mathbf{E}(X_\varphi) = p^{|E_H|}$. The number of choices for φ is $n(n-1)\cdots(n-|V_H|+1) = \dfrac{n!}{(n-|V_H|)!}$.

(b) The number of possible edges in a set of $|V_H|$ vertices is $\binom{|V_H|}{2}$. Call this number H_2. The answer is $\dfrac{n!}{(n-|V_H|)!} p^{|E_H|} (1-p)^{H_2 - |E_H|}$.

(c) Suppose $V_H = \{a, b, c\}$ and that $\{p, q, r\}$ are the vertices of a triangle in a random graph. The triangle formed by $\{p, q, r\}$ in counted once in Example 6.13. In part (a) it is counted 6 times because there are 6 injections $\varphi : \{a, b, c\} \to \{p, q, r\}$.

6.5.3 (a) The probability of a cycle is the probability of the union of the sets $\mathcal{G}_C$. The probability of the union of sets, is less than or equal to the sum of their separate probabilities; that is, $\Pr(A \cup B \cup \cdots) \le \Pr(A) + \Pr(B) + \cdots$.

(b) The denominator is $|\mathcal{G}(n,k)|$. The numerator counts graphs as follows. There are $(c-1)!$ directed cycles. Since each cycle can be made directed in two ways, there are $(c-1)!/2$ cycles. Since we have used up c edges making the cycle, we must choose $k-c$ edges from the remaining $N-c$ unused edges.

(c) Collect terms in (a) according to $c = |C|$ and use (b). There are $\binom{n}{c}$ c-subsets of $\underline{n}$.

(d) The left side comes from writing $\binom{x}{m} = \frac{x(x-1)\cdots(x-m+1)}{m!}$ and doing some algebra. The inequality comes from $\frac{k!}{(k-c)!} < k^c$ and $\frac{x-j}{y-j} < \frac{x}{y}$ when $y > x \ge j$.

6.5.4 (a) We need k edges. Since each has probability p and they are independent in $\mathcal{G}_p(n)$, the answer is p^k.

(b) There are less than n^k/k possible cycles. By (a), each has probability less than p^k.

(c) By (b), the probability of a cycle is less than

$$\sum_{k\ge 3}(pn)^k/k \;<\; \sum_{k\ge 3}(pn)^k/3 \;=\; \frac{(pn)^3}{3(1-pn)} \;<\; (pn)^3.$$

6.5.5 (a) Let T contain a close to half the vertices as possible. If $|V| = 2n$, $|T| = n$ and $|V-T| = n$. Since G contains all edges, this choice of T gives us a bipartite subgraph with n^2 edges. When $|V| = 2n+1$, we take $|T| = n$ and $|V-T| = n+1$, obtaining a bipartite subgraph with $n(n+1)$ edges.

(b) The example bound is $|E|/2$ and $|E| = |\mathcal{P}_2(V)| = |V|(|V|-1)/2$. For $|V| = 2n$, we have $|E|/2 = n(2n-1)/2 = n^2 - n/2$. Hence the bound is off by $n/2$. This may sound large, but the relative error is small: Since $(n^2-n/2)/n^2 = 1 - 1/2n$, the relative error is $1/|V|$. We omit similar calculations for $|V| = 2n+1$.

(c) The idea is to construct the largest possible complete graph and then add edges in any manner whatsoever. Let m be the largest integer such that $k \ge \binom{m}{2}$, choose $S \subseteq V$ with $|S| = m$, construct a complete graph on m vertices using $\binom{m}{2}$ edges, and insert the remaining $k - \binom{m}{2}$ edges in any manner to form a simple graph $G(V,E)$. By (a), the number of edges in a bipartite subgraph of the complete graph on T has at least $(m/2)^2 - m$ edges for some constant C Since m is as large as possible, $k < \binom{m+1}{2} < \frac{(m+1)^2}{2}$. Thus $m+1 > \sqrt{2k}$. Also, since $k \ge \binom{m}{2} > \frac{(m-1)^2}{2}$, $m-1 < \sqrt{2k}$. Hence the number of edges in bipartite subgraph is at least

$$(m/2)^2 - m \;>\; \frac{\left(\sqrt{2k}\,-1\right)^2}{4} - \sqrt{2k} - 1,$$

Which equals k minus terms involving $k^{1/2}$ and constants.

(d) Call the colors 1,2,3. Let V_i be the set of vertices colored with color i and let $E_{i,j}$ be the set of edges in G that connect vertices in V_i to vertices in V_j. Since $|E| = |E_{0,1}| + |E_{0,2}| + |E_{1,2}|$, at least one of $|E_{i,j}|$ is at most $|E|/3$. Suppose it is $E_{1,2}$. The bipartite subgraph whose edges connect vertices in V_0 to vertices in $V_1 \cup V_2$ contains $E - |E_{1,2}| \ge 2|E|/3$ edges.

Section 6.6

6.6.1. The left column gives the input and the top row the states.

	0	1	2	3	4
0	0	2	4	1	3
1	1	3	0	2	4

6.6.2 (a) and (b) The digraph can be drawn from the transition table. There will be 3 states, 0,1,2, corresponding to the remainder after dividing the number by 3. Thus the starting and accepting state are both 0. Here's the transition table.

	0	1
0	0	1
1	2	0
2	1	2

(c) Here's the transition table. The starting and accepting states are both 0.

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	0	1	2	0	1	2	0
1	2	0	1	2	0	1	2	0	1	2
2	1	2	0	1	2	0	1	2	0	1

(d) What follows is probably the easiest and most natural way to think about the problem, but there is a more general idea that is found in the next part. Since 10 has a remainder of 1 when divided by 3, the remainder of $a_k 10^k$ when divided by 3 is the same as the remainder of a_k. Thus processing a number from left to right will be the same as from right to left. Consequently the previous transition table works.

(e) If we attempt to use the idea of the previous part, we see that the remainder of dividing 2^k by 3 depends on whether k is odd or even. This suggests that our state consist of two parts, one for the parity of k and one for the remainder so far. There is another way to think about this that works in general. Suppose that we are working in base b and looking at divisibility by d. Also suppose that B is such that bB has remainder 1 when divided by d. Note that a number N is divisible by d if and only if NB^n is divisible by d and that

$$B^n(a_n b^n + \cdots + a_1 b^1 + a_0 b^0) \;=\; a_0 B^n + a_1 B^{n-1}(bB) + \cdots + a_n B^0 (bB)^n.$$

Using the fact the $(bB)^k$ has remainder 1 when divided by d, This number has the same remainder as $a_0 B^n + \cdots + a_n B^0$. This means that our original number is divisible by d if and only if the number we get by switching right and left and changing to base B is divisible by d. With $b = 10$ and $d = 3$, we can take $B = 10$ since dividing $bB = 100$ by 3 gives remainder 1. Hence the solution for left to right works for right to left. As a further example, suppose the base is still 10 but now $d = 7$. Since 50 has a remainder of 1, we can set $B = 5$.

6.6.3. The states are 0, O1, E1 and R. In state 0, a zero has just been seen; in O1, an odd number of ones; in E1, an even number. The start state is 0 and the accepting states are 0 and O1. The state R is entered when we are in E1 and see a 0. Thereafter, R always steps to R regardless of input. You should be able to finish the machine.

	σ	$\bullet$	E	δ	comments on state
b	s1	sd	z	1a	starting
s1	z	sd	z	1a	part 1 sign seen
1a	z	2	e	1a	part 1 digits seen; accepting
sd	z	z	z	1b	decimal seen, no digits yet
1b	z	z	e	1b	part 1 after decimal; accepting
e	s2	z	z	2	E seen
s2	z	z	z	2	part 2 sign seen
2	z	z	z	2	part 2 digits seen; accepting
z	z	z	z	z	error seen

Figure S.6.2 The transition table for a finite automaton that recognizes floating point numbers, the possible inputs are sign (σ), decimal point ($\bullet$), digit (δ) and exponent symbol (E). The comments explain the states.

6.6.4. If you understand what this automaton is recognizing and the significance of the states, it makes the problem easier. It is looking for strings of digits which may have a sign to begin with. The state b corresponds to having seen nothing, s to having seen a sign, d a digit and z something illegal.

(b) It recognizes all nonempty strings of digits with an optional sign at the start. Thus it would not recognize the string "+".

(c) It recognizes all nonempty strings that consist of an optional sign followed by digits. Thus it would recognize the string "+".

6.6.5. In our input, we let δ stand for any digit, since the transition is independent of which digit it is. Similarly, σ stands for any sign. There is a bit of ambiguity as to whether the integer after the E must have a sign. We assume not. The automaton contains three states that can transit to themselves: recognizing digits before a decimal, recognizing digits after a decimal and recognizing digits after the E. We call them 1a, 1b and 2. There is a bit of complication because of the need to assure digits in the first part and, if it is present, in the second part. The transition table is given in Figure S.6.2.

6.6.6 (a) Our states will be 0, 1, p, and i, where p indicates that we have just seen what could be an isolated 1 and i indicates that we have seen an isolated one. The start state is 0 and the accepting states are p and i. Here's the transition table.

	0	1
0	0	p
1	0	1
p	i	1
i	i	i

(b) We can join together what look roughly like two copies of the previous machine. The states in the second one are postfixed with an r and are used to look for a second isolated one. The

	5	10	25	A	B	C	R
0	5	10	25	0	0	0	0
5	10	15	30	5	5	5	0, R5
10	15	20	10, R25	10	10	10	0, R10
15	20	25	15, R25	0, A0	15	15	0, R15
20	25	30	20, R25	0, A5	0, B0	20	0, R20
25	30	25, R10	25, R25	0, A10	0, B5	0, C0	0, R25
30	30, R5	30, R10	30, R25	0, A15	0, B10	0, C5	0, R30

Figure S.6.3 The transitions and outputs for an automaton that behaves like a vending machine. The state is the amount of mone held and the input is either money, a purchase choice (A, B, C) or a refund request (R).

accepting states are p, 0r, and 1r.

	0	1
0	0	p
1	0	1
p	0r	1

	0	1
0r	0r	pr
1r	0r	1r
pr	ir	1r
ir	ir	ir

6.6.7 (a) We need states that keep track of how much money is held by the machine. This leads us to states named $0, 5, \ldots, 30$. The output of the machine will be indicated by An, Bn, Cn and n, where n indicates the amount of money returned and A, B and C indicate the item delivered. There may be no output. The start state is 0.

(b) See Figure S.6.3.

6.6.8 (a) Let

$$\mathcal{M} \times \mathcal{M}' = (S \times S',\ I,\ f \times f',\ (s_o, s_o'),\ A \times A')$$

where $(f \times f')(s, s', i) = (f(s,i),\ f'(s', i))$.

(b) There is an edge from (s, s') to (t, t') if and only if there is an $i \in I$ such that $f(s,i) = t$ and $f'(s', i) = t'$. The edge is associated with the input i.

(c) A number is divisible by 15 if and only if it is divisible by 5 and divisible by 3. If the two given machines are called $\mathcal{M}$ and $\mathcal{M}'$, we simply look at $\mathcal{M} \times \mathcal{M}'$.

(d) The machine in the previous part can be used; however, the accepting states must be those for which either remainder is zero.

Section 7.1

7.1.1. $\mathcal{A}(m)$ (note m, not n) is the statement of the rank formula. The inductive step and use of the inductive hypothesis are clearly indicated in the proof.

7.1.2. $\mathcal{A}(n)$ is the claim that $D_1, D_2, \ldots, D_n$ has been chosen by the greedy algorithm and is part of the correct path. The inductive hypothesis is used in the assumption that $D_1, \ldots, D_{i-1}, D'_i$ is part of the correct path for some D'_i.

7.1.3. Let $\mathcal{A}(k)$ be the assertion that the coefficient of $y_1^{m_1} \cdots y_k^{m_k}$ in $(y_1+\cdots+y_k)^n$ is $n!/m_1!\cdots m_k!$ if $n = m_1 + \cdots + m_k$ and 0 otherwise. $\mathcal{A}(1)$ is trivial. We follow the hint for the induction step. Let $x = y_1 + \cdots + y_{k-1}$. By the binomial theorem, the coefficient of $x^m y_k^{m_k}$ in $(x + y_k)^n$ is $n!/m!m_k!$ if $n = m + m_k$ and 0 otherwise. By the induction hypothesis, the coefficient of $y_1^{m_1} \cdots y_{k-1}^{m_{k-1}}$ in x^m is $m!/m_1!\cdots m_{k-1}!$ if $m = m_1 + \cdots + m_{k-1}$ and zero otherwise. Combining these results we see that the coefficient of $y_1^{m_1} \cdots y_k^{m_k}$ in $(y_1 + \cdots + y_k)^n$ is

$$\frac{n!}{m!\, m_k!}\,\frac{m!}{m_1!\cdots m_{k-1}!}$$

if $n = m_1 + \cdots + m_k$ and 0 otherwise.

7.1.4 (a) Since (ii) starts at $n = 2$, the case $D_1 = 1D_0 + (-1)^1$ must be proved directly. That's easy. For $n \geq 2$ we have, with a bit of trickiness,

$$\begin{aligned} D_n &= nD_{n-1} + (-1)^n = (n-1)D_{n-1} + (-1)^n + D_{n-1} && \text{by (i) at } n \\ &= (n-1)D_{n-1} + (-1)^n + (n-1)D_{n-2} + (-1)^{n-1} && \text{by (i) at } n-1 \\ &= (n-1)(D_{n-1} + D_{n-2}). \end{aligned}$$

(b) Let $\mathcal{A}(n)$ be the claim that $D_n = nD_{n-1} + (-1)^n$. $\mathcal{A}(1)$ is easily checked. Now for the inductive step.

$$\begin{aligned} D_n &= (n-1)(D_{n-1} + D_{n-2}) = nD_{n-1} + (n-1)D_{n-2} - D_{n-1} \\ &= nD_{n-1} + (n-1)D_{n-2} - ((n-1)D_{n-2} + (-1)^{n-1}) && \text{by } \mathcal{A}(n-1) \\ &= nD_{n-1} + (-1)^n. \end{aligned}$$

(c) Let $\mathcal{A}(n)$ be the desired equation. It is easy to verify $\mathcal{A}(0)$. Now for the induction step when $n \geq 1$. We have

$$D_n \;=\; nD_{n-1} + (-1)^n \;=\; n\,(n-1)! \sum_{k=0}^{n-1} \frac{(-1)^k}{k!} + n!\frac{(-1)^n}{n!} \;=\; n! \sum_{k=0}^{n} \frac{(-1)^k}{k!},$$

where the second equality used $\mathcal{A}(n-1)$.

(d) Using (iii) twice we have

$$D_n \;=\; n! \sum_{k=0}^{n} \frac{(-1)^k}{k!} \;=\; n! \sum_{k=0}^{n-1} \frac{(-1)^k}{k!} + (-1)^n \;=\; nD_{n-1} + (-1)^n.$$

7.1.5 (a) $x'_1x'_2 + x'_1x_2 \;=\; x'_1$.

(b) $x'_1x_2 + x_1x'_2$.

(c) $x'_1x'_2x_3 + x'_1x_2x_3 + x_1x'_2x'_3 + x_1x_2x'_3 \;=\; x'_1x_3 + x_1x'_3$.

(d) $x'_1x'_2x_3 + x'_1x_2x'_3 + x'_1x_2x_3 + x_1x'_2x'_3 \;=\; x'_1x_2 + x'_1x_3 + x_1x'_2x'_3$.

7.1.6. This can be done in various ways. One possible approach is to use the distributive law to expand the products. Another is to use the method that was used in proving the theorem.

(a) By the distributive law: $x_1x_2 + x_1x_4 + x_3x_4$.
By the proof of the theorem:

$$\begin{aligned}(x_1+x_3)(x_2+x_4) &= (x_1+x_3)(x_2+0)x_4' + (x_1+x_3)(x_2+1)x_4\\ &= (x_1+x_3)x_2x_4' + (x_1+x_3)x_2'x_4\\ &= (x_1+0)x_3'x_2x_4' + (x_1+1)x_3x_2x_4' + (x_1+0)x_3'x_2'x_4 + (x_1+1)x_3x_2'x_4\\ &= x_1x_3'x_2x_4' + x_1'x_3x_2x_4' + x_1x_3'x_2'x_4 + x_1'x_3x_2'x_4.\end{aligned}$$

(b) By the distributive law, remembering that $xx = x$ and $x + xy = x$:

$$(x_1 + x_2x_3)(x_2 + x_3) = x_1x_2 + x_1x_3 + x_2x_3.$$

There are a variety of forms with more terms.

7.1.7. If you are familiar with de Morgan's laws for complementation, you can ignore the hint and give a simple proof as follows. By Example 7.3, one can express f' in disjunctive form: $f' = M_1 + M_2 + \cdots$. Now $f = (f')' = M_1'M_2'\cdots$ by de Morgan's law and, if $M_i = y_1y_2\cdots$, then $M_i' = y_1' + y_2' + \cdots$ by de Morgan's law.

To follow the hint, replace (7.5) with

$$f(x_1,\ldots,x_n) = \left(g_1(x_1,\ldots,x_{n-1}) + x_n'\right)\left(g_0(x_1,\ldots,x_{n-1}) + x_n\right)$$

and practically copy the proof in Example 7.3.

7.1.9. We can induct on either k or n. It doesn't matter which we choose since the formula we have to prove is symmetric in n and k. We'll induct on n. The given formula is $\mathcal{A}(n)$. For $n = 0$, the formula becomes $F_{k+1} = F_{k+1}$, which is true.

$$\begin{aligned}F_{n+k+1} &= F_{(n-1)+(k+1)+1} && \text{using the hint}\\ &= F_nF_{k+2} + F_{n-1}F_{k+1} && \text{by } \mathcal{A}(n-1)\\ &= F_n(F_{k+1}+F_k) + F_{n-1}F_{k+1} && \text{by definition of } F_{k+2}\\ &= (F_n+F_{n-1})F_{k+1} + F_nF_k && \text{by rearranging}\\ &= F_{n+1}F_{k+1} + F_nF_k && \text{by definition of } F_{n+1}.\end{aligned}$$

Section 7.2

7.2.1. Given that p and q are positive integers, it does not follow that p' and q' are positive integers. (For example, let $p = 1$.) Thus $\mathcal{A}(n-1)$ may not apply.

7.2.2. The last sentence in the proof is quite vague: It does not explain how one is going to actually use the drawing. Any attempt to make the sentence more precise is bound to fail because one cannot carry out the idea described there.

7.2.3. You may object that the induction has not been clearly phrased, but this can be overcome: Let I be the set of interesting positive integers and let $\mathcal{A}(n)$ be the assertion $n \in I$. If $\mathcal{A}(1)$ is false, then even 1 is not interesting, which is interesting. The inductive step is as given in the problem: If $\mathcal{A}(n)$ is false, then since $\mathcal{A}(k)$ is true for all $k < n$, n is the smallest uninteresting number, which is interesting.

Then what *is* wrong? It is unclear what "interesting" means, so the set of interesting positive integers is not a well defined concept. Proofs based on foggy concepts are always suspect.

7.2.4. There is not reduction of the problem to a simpler case. We could overcome this by assigning numbers to the students and making sure that a person always asks someone with a lower number, but then student number 1 would have no one in the class to turn to.

Section 7.3

7.3.1 (a) We must compare as long as both lists have items left in them. After all items have been removed from one list, what remains can simply be appended to what has been sorted. All items will be removed from one list the quickest if each comparison results in removing an item from the shorter list. Thus we need at least $\min(k_1, k_2)$ comparisons.

On the other hand, suppose we have $k_1 + k_2$ items and the smallest ones are in the shorter list. In this case, all the items are removed from the shorter list and none from the longer in the first $\min(k_1, k_2)$ comparisons, so we have achieved the minimum.

(b) Here's the code. Note that the two lists have lengths m and $n-m$ and that $\min(m, n-m) = m$ because $m \leq n/2$.

```
Procedure c(n)
    c = 0
    If (n = 1), then Return c
    Let m be n/2 with remainder discarded
    c = c + c(m)
    c = c + c(n − m)
    c = c + m
    Return c
End
```

(c) We have $c(2^0) = 0$ and $c(2^{k+1}) = 2c(2^k) + 2^k$ for $k \geq 0$. The first few values are

$$c(2^0) = 0, \quad c(2^1) = 2^0, \quad c(2^2) = 2 \times 2^1, \quad c(2^3) = 3 \times 2^2, \quad c(2^4) = 4 \times 2^3.$$

This may be enough to suggest the pattern $c(2^k) = k \times 2^{k-1}$; if not, you can compute more values until the pattern becomes clear.

We prove it by induction. The conjecture $c(2^k) = k \times 2^{k-1}$ is the induction assumption.

For $k = 0$, we have $c(2^0) = 0$, and this is what the formula gives. For $k > 0$, we use the recursion to reduce k and then use the induction assumption:

$$c(2^k) \;=\; 2c(2^{k-1}) + 2^{k-1} \;=\; 2 \times (k-1) \times 2^{k-2} + 2^{k-1} \;=\; k \times 2^{k-1},$$

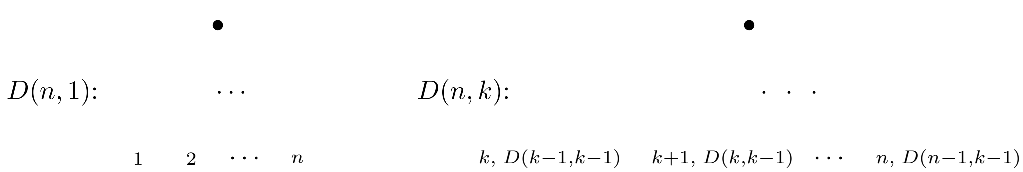
which completes the proof.

When k is large,

$$\frac{c(2^k)}{C(2^k)} \;=\; \frac{k \times 2^{k-1}}{(k-1)2^k + 1} \;=\; \frac{k/2}{k-1+2^{-k}} \;\sim\; 1/2.$$

This shows that the best case and worst case differ by about a factor of 2, which is not very large.

7.3.2. We have



7.3.3. Here is code for computing the number of moves.

```
Procedure M(n)
    M = 0
    If (n = 1), then Return M
    Let m be n/2 with remainder discarded
    M = M + M(m)
    M = M + M(n − m)
    M = M + n
    Return M
End
```

This gives us the recursion $M(2^k) = 2M(2^{k-1}) + 2^k$ for $k > 0$ and $M(2^0) = 0$. The first few values are

$$M(2^0) = 0, \quad M(2^1) = 2^1, \quad M(2^2) = 2 \times 2^2, \quad M(2^3) = 3 \times 2^3, \quad M(2^4) = 4 \times 2^4.$$

Thus we guess $M(2^k) = k2^k$, which can be proved by induction.

7.3.4. We specified that `Find` was to always report a counterfeit coin, but when we used it recursively we assumed that it could also report that there was no counterfeit coin. To allow for this, we must alter "`Else report C.`" `C` must be compared with some other coin to determine whether or not it is counterfeit. The corrected algorithm requires $n-1$ weighings, which is very poor since only about $\log_2 n$ weighings are needed.

7.3.5 (a) Here's one possible procedure. Note that the remainder must be printed out *after* the recursive call to get the digits in the proper order. Also note that one must be careful about zero: A string of zeroes should be avoided, but a number which is zero should be printed.

```
OUT(m)
     If m < 0, then
          Print ‘‘−’’
          Set m = −m
     End if
     Let q and 0 ≤ r ≤ 9 be determined by m = 10q + r
     If q > 0, then OUT(q)
     Print r
End
```

(b) Single digits

(c) When OUT calls itself, it passes an argument that is smaller in magnitude than the one it received, thus OUT(m) must terminate after at most $|m|$ calls.

7.3.6 (a)

```
DSUM(n)
     If n = 0, Return 0.
     Let q and 0 ≤ r ≤ 9 be determined by n = 10q + r.
     Return DSUM(q) + r.
End
```

(b) Zero

(c) Same as previous exercise.

7.3.7. The description for $k = 1$ is on the left and that for $k > 1$ is on the right:

$$\underline{n^1} \qquad\qquad \underline{n^k}$$

$$1 \quad \cdots \quad n \qquad\qquad k,\ \underline{\underline{k-1}^{k-1}} \quad \cdots \quad n,\ \underline{\underline{n-1}^{k-1}}$$

7.3.8. We will not draw the trees. The moves for $n = 2$ are $S\xrightarrow{1}E$, $S\xrightarrow{2}G$ and $E\xrightarrow{1}G$. The moves for $n = 4$ are (reading row by row)

$S\xrightarrow{1}E$ $S\xrightarrow{2}G$ $E\xrightarrow{1}G$ $S\xrightarrow{3}E$ $G\xrightarrow{1}S$ $G\xrightarrow{2}E$ $S\xrightarrow{1}E$ $S\xrightarrow{4}G$
$E\xrightarrow{1}G$ $E\xrightarrow{2}S$ $G\xrightarrow{1}S$ $E\xrightarrow{3}G$ $S\xrightarrow{1}E$ $S\xrightarrow{2}G$ $E\xrightarrow{1}G$

7.3.9 (a) Let $\mathcal{A}(n)$ be the assertion “H(n, S, E, G) takes the least number of moves.” Clearly $\mathcal{A}(1)$ is true since only one move is required. We now prove $\mathcal{A}(n)$. Note that to do $S\xrightarrow{n}G$ we must first move all the other washers to pole E. They can be stacked only one way on pole E, so moving the washers from S to E requires using a solution to the Tower of Hanoi problem for $n-1$ washers. By $\mathcal{A}(n-1)$, this is done in the least number of moves by H$(n-1, S, G, E)$. Similarly, H$(n-1, E, S, G)$ moves these washers to G in the least number of moves.

(b) Simply replace H$(m, \ldots)$ with $S(m)$ and replace a move with a 1 and adjust the code a bit to get

```
Procedure S(n)
    If (n = 1)  Return 1.
    M = 0
    M = M + S(n − 1)
    M = M + 1
    M = M + S(n − 1)
    Return M
End
```

The recursion is $S(1) = 1$ and $S(n) = 2S(n-1) + 1$ when $n > 1$.

(c) The values are 1, 3, 7, 15, 31, 63, 127.

(d) Let $\mathcal{A}(n)$ be "$S(n) = 2^n - 1$." $\mathcal{A}(1)$ asserts that $S(1) = 1$, which is true. By the recursion and then the induction hypothesis we have

$$S(n) \;=\; 2S(n-1)+1 \;=\; 2(2^{n-1}-1)+1 \;=\; 2^n - 1.$$

(e) By studying the binary form of k and the washer moved for small n (such as $n = 4$) you could discover the following rule.

If $k = \cdots b_3b_2b_1$ is the binary representation of k, $b_j = 1$, and $b_i = 0$ for all $i < j$, then washer j is moved.

(This simply says that b_j is the lowest nonzero binary digit.) No proof was requested, but here's one. Let $\mathcal{A}(n)$ be the claim for $\mathtt{H}(n, \ldots)$. $\mathcal{A}(1)$ is trivial. We now prove $\mathcal{A}(n)$. If $k < 2^{n-1}$, it follows from $S(m)$ that $\mathtt{H}(n-1, \ldots)$ is being called and $\mathcal{A}(n-1)$ applies. If $k = 2^{n-1}$, then we are executing $S \xrightarrow{n} G$ and so this case is verified. Finally, if $2^{n-1} < k < 2^n$, then $\mathtt{H}(n-1, \ldots)$ is being executed at step $k - 2^{n-1}$, which differs from k only in the loss of its leftmost binary bit.

(f) Suppose that we are looking at move $k = \cdots b_3b_2b_1$ and that washer j is being moved. (That means b_j is the rightmost nonzero bit.) You should be able to see that this is move number $\cdots b_{j+2}b_{j+1} = (k - 2^{j-1})/2^j$ for the washer. Call this number k'. To determine source and destination, we must study move patterns.

The pattern of moves for a washer is either

$$P_0\colon S \to G \to E \to S \to G \to E \to \cdots \text{ repeating or}$$
$$P_1\colon S \to E \to G \to S \to E \to G \to \cdots \text{ repeating.}$$

Which washer uses which pattern? Consider washer j it is easily verified that it is moved a total of 2^{n-j} times, after which time it must be at G. A washer following P_i is at G only after move numbers of the form $3t + i + 1$ for some t. Thus $i + i$ is the remainder when 2^{n-j} is divided by 3. The remainder is 1 if $n-j$ is even and 0 otherwise. Thus washer j follows pattern P_i where i and $n - j$ have the same parity. If we look at the remainder after dividing k' by 3, we can see what the source and destination are by looking at the start of P_i. For those of you familiar with congruences, the remainder is congruent to $(-1)^j k + 1$ modulo 3.

7.3.10. We are not actually reducing to a simpler problem because we cannot ignore the presence of washer k on a pole and move larger washers on top of it.

In the text, we ignored the presence of the largest washer. This is actually reducing to a simpler problem because we can pile other washers on top of it as if it were not there.

7.3.11 (a) We have

$$H^*(n,S,E,G)$$

$$H^*(n-1,S,E,G) \qquad S \xrightarrow{n} E \qquad H^*(n-1,G,E,S) \qquad E \xrightarrow{n} G \qquad H^*(n-1,S,E,G)$$

(b) The initial condition is $h_1^* = 2$. For $n > 1$ we have $h_n^* \; = \; 3h_{n-1}^* + 2$. Alternatively, $h_0^* = 0$ and, for $n > 0$, $h_n^* \; = \; 3h_{n-1}^* + 2$.

(c) The general solution is $h_n^* \; = \; 3^n - 1$. To prove it, use induction. First, it is correct for $n = 0$. Then, for $n > 0$,

$$h_n^* \; = \; 3h_{n-1}^* + 2 \; = \; 3(3^{n-1} - 1) + 2 \; = \; 3^n - 1.$$

7.3.12 (a) You may not have taken care of all the initial conditions in your code if you didn't state the recursion carefully. We'll use

$$S(n,k) \; = \; S(n-1,k-1) + kS(n-1,k) \qquad \text{when} \quad n > 0 \text{ and } k > 1$$

with the initial conditions $S(0,k) = 0$ for $k > 0$ and $S(n,1) = 1$ for $n > 0$.

It will be useful to define two operations. Let $\mathcal{P}$ be a collection of partitions of a set not containing t. Define $\texttt{Add}(\mathcal{P},t)$ to be the collection of partitions $P \cup \{t\}$ where $P \in \mathcal{P}$; that is, the result of adding the block $\{t\}$ to each partition in P. Define $\texttt{Ins}(\mathcal{P},t)$ to be the collection of partitions $(P - B) \cup \{B \cup \{t\}\}$ where $B \in P \in \mathcal{P}$; that is, the result of adding t to one of the blocks of each partition in P. Note that if P has k blocks, then t is added to each in turn producing k new partitions.

```
S(T,k)
    /* Do the S(0,k) case. */
    If T = ∅, Return ∅.
    /* Do the S(n,1) case. */
    If k = 1, then Return {T}.
    Select t ∈ T and let U = T − {t}.
    Return Add(S(U,k−1),t) ∪ Ins(S(U,k),t)
End
```

(b) The three cases show here ar $T = \emptyset$, $k = 0$ and the rest; i.e., $T \neq \emptyset$ and $k > 0$. As in the code, $U = T - \{t\}$ for some $t \in T$.

$$\texttt{S}(\emptyset,k) \qquad \texttt{S}(T,1) \qquad \qquad \texttt{S}(T,k)$$

$$\emptyset \qquad \{T\} \qquad \texttt{Add}(\texttt{S}(U,k-1),t) \qquad \texttt{Ins}(\texttt{S}(U,k),t)$$

7.3.13 (b) Induct on n. It is true for $n = 1$. If $n > 1$, $a_2, \ldots, a_n \in G(k_2, \ldots, k_n)$ by the induction hypothesis. Thus $a_1, a_2, \ldots, a_n$ is in a_1, H and $a_1, R(H)$.

(c) Induct on n. It is true for $n = 1$. Suppose $n > 1$ and let the adjacent leaves be $b_1, \ldots, b_n$ and $c_1, \ldots, c_n$, with c following b. If $b_1 = c_1$, then apply the induction hypothesis to $G(k_2, \ldots, k_n)$ and the sequences $b_2, \ldots, b_n$ and $c_2, \ldots, c_n$. If $b_1 \neq c_1$, it follows from the local description that $c_1 = b_1 + 1$, that $b_2, \ldots, b_n$ is the rightmost leaf in H (or $R(H)$) and that $c_2, \ldots, c_n$ is the leftmost leaf in $R(H)$ (or H, respectively). In either case, $b_2, \ldots, b_n$ and $c_2, \ldots, c_n$ are equal because they are the same leaf of H.

(d) Let $R_n(\alpha)$ be the rank of $\alpha_1, \ldots, \alpha_n$. Clearly $R_1(\alpha) = \alpha_1 - 1$. If $n > 1$ and $\alpha_1 = 1$, then $R_n(\alpha) = R_{n-1}(\alpha_2, \ldots, \alpha_n)$. If $n > 1$ and $\alpha_1 = 2$, then $R_n(\alpha) = 2^n - 1 - R_{n-1}(\alpha_2, \ldots, \alpha_n)$. Letting $x_i = \alpha_i - 1$, we have $R_n(\alpha) = (2^n - 1)x_1 + (-1)^{x_1} R_{n-1}(\alpha_2, \ldots, \alpha_n)$ and so

$$R_n(\alpha) = (2^n-1)x_1+(-1)^{x_1}(2^{n-1}-1)x_2+(-1)^{x_1+x_2}(2^{n-2}-1)x_3+\cdots+(-1)^{x_1+x_2+\cdots+x_{n-1}}x_n.$$

(e) If you got this, congratulations. Let j be as large as possible so that $\alpha_1, \ldots, \alpha_j$ contains an even number of 2's. Change α_j. (Note: If $j = 0$, $\alpha = 2, 1, \ldots, 1$, the sequence of highest rank, and it has no successor.)

7.3.14. Your answer here will depend on exactly how you set up your procedures. In the procedures we have written, information on where to return, perhaps some temporary storage for compiler generated variables, and the following variables will all go onto the stack.

1. The integers m, q and r.
2. The integers n, q and r.
4. The integer n.
7. The sets T and U, the integer k and the element t. This pseudocode is rather far from actual code in many languages because the procedure returns a set whose elements are partitions of a set. There will undoubtedly be some storage associated with this, perhaps in the form of a linked list of pointers. As a result of replacing the pseudocode with code, we would probably create a few additional variables that are pointers.

Section 7.4

7.4.1. Let $M(n)$ be the minimum number of multiplications needed to compute x^n. We leave it to you to verify the following table for $n \le 9$

n	2	3	4	5	6	7	8	9	15	21	47	49
$M(n)$	1	2	2	3	3	4	3	4	5	6	8	7

Since $15 = 3 \times 5$, it follows that $M(15) \le M(3) + M(5) = 5$. Likewise, $M(21) \le M(3) + M(7) = 6$. Since the binary form of 49 is 110001_2, $M(49) \le 7$. Since $47 = 101111_2$, we have $M(47) \le 9$, but we can do better. Using $47 = 2 \times 23 + 1$, gives $M(47) \le M(23) + 2$, which we leave for you to work out. A better approach is given by $47 = 5 \times 9 + 2$. Since x^2 is computed on the way to finding x^5, it is already available and so $M(49) \le M(5) + M(9) + 1 = 8$. It turns out that these are minimal, but we will not prove that.

7.4.2 (a) Starting with F_0, the values are 0, 1, 1, 2, 3, 5, 8 and 13.

(b) The equations follow immediately from the recursion and the definition of M. To compute F_n, use the ideas in Example 7.21 to calculate $P = M^{n-1}$ rapidly. Then $F_n = (P\vec{v}_0)_2 = P_{2,2}$.

Another approach is to use linear algebra. If λ_1, λ_2 are the eigenvalues of M, then $M = Q \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} Q^{-1}$ for some matrix Q and so $M^n = Q \begin{pmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{pmatrix} Q^{-1}$.

(c) Write $R = M^n$. Since $(F_n, F_{n+1})^{\mathrm{t}} = R(0,1)^{\mathrm{t}}$, $F_n = R_{1,2}$ and $F_{n+1} = R_{2,2}$. Since

$$(F_{n+1}, F_{n+2})^{\mathrm{t}} = R\vec{v}_1 = R(1,1)^{\mathrm{t}},$$

$F_{n+1} = R_{1,1} + R_{1,2}$ and $F_{n+2} = R_{2,1} + R_{2,2}$. Subtract the two earlier equations from these and use the recursion for the Fibonacci numbers.

(d) This follows from the previous part and, for F_{2n}, the rearranged recursion $F_n = F_{n+1} - F_{n-1}$.

7.4.3. Let $\vec{v}_n$ be the transpose of $(a_n, \ldots, a_{n+k-1})$. Then $\vec{v}_n = M\vec{v}_{n-1}$ where

$$M = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_k & a_{k-1} & a_{k-2} & \cdots & a_1 \end{pmatrix}.$$

7.4.4 (a) Divide the coins into three nearly equal piles, P_1, P_2 and P_3 in such a way that the first two piles have an equal number of coins. Compare the first two piles in the scales. If they differ in weight, the counterfeit coin is in the lighter pile. If they have the same weight, the counterfeit coin is in P_3.

(b) There are n possibilities for the counterfeit coin. Construct the decision tree for the algorithm, labeling the leaves with the identity of the counterfeit coin. Thus, there must be at least n leaves. Since weighing has three possible outcomes (equal, heavier and lighter), each vertex in the tree has at most three sons. In the best possible case, each nonleaf vertex will have three sons with at most one exception which has two sons. Also, the sons of this exception are leaves and no leaves differ in height by more than one. (These facts can be proved as they are for sorting.) The rest of the argument is like the proof of the sorting theorem.

(c) Proceed as in the known case; however, compare P_1 with P_2 and compare P_1 with P_3. This determines if the counterfeit is lighter or heavier and which of the three piles it is in. The rest of the weighings now proceed as in the case where the counterfeit was known to be lighter. To see that the relative weight of the counterfeit is determined simply consider the possible cases as shown in the table, where w_i is the weight of P_i and an entry i, j indicates that the counterfeit is in pile P_i and it is lighter or heavier according as j =L or j =H.

	$w_1 < w_3$	$w_1 = w_3$	$w_1 > w_3$
$w_1 < w_2$	1,L	2,H	never
$w_1 = w_2$	3,H	never	3,L
$w_1 > w_2$	never	2,L	1,H

(d) One possibility is to again divide the coins into three piles. If $w_1 \neq w_2$, the heavier pile contains no counterfeits and so can be removed. Merge the other two piles and start again. If $w_1 = w_2$, then either $w_1 > w_3$ or $w_1 < w_3$. In the former case, both counterfeit coins are in P_3, so we can start again with P_3. If $w_1 < w_3$, one coin is in each of P_1 and P_2 and they can be searched separately by our earlier strategy.

7.4.5. Finding a maximum of n items can be done in $\Theta(n)$, so it's the computation of all the different $F(v)$ values is the problem. Thus we could compute the values of F separately from finding the maximum. However, since it's convenient to compute the maximum while we're computing the values of F, we'll do it.

The root r of T has two sons, say s_L and s_R. Observe that the answer for the tree rooted at r must be either the answer for the tree rooted at s_L or the answer for the tree rooted at s_R or $F(r)$. Also

$$F(r) = f(r) + F(s_L) + F(s_R).$$

Here's an algorithm that carries out this idea.

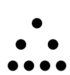
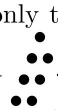
```
/* r is the root of the tree in what follows. */

Procedure BestSum(r)
     Call Recur(r, Fvalue, best)
     Return best
End

Procedure Recur(r, F, best)
     If r is a leaf then
          F = f(r)
          best = f(r)
     Else
          Let s_L and s_R be the sons of r.
          Recur(s_L, F_L, b_L)
          Recur(s_R, F_R, b_R)
          F = f(r) + F_L + F_R
          best = max(F, b_L, b_R)
     End if
     Return
End
```

Since $f(r)$ is only used once, the running time of this algorithm is $\Theta(n)$ for n vertices, an improvement over $\Theta(n \ln n)$.

7.4.6. Now instead of an interval that is divided into two pieces, a fast algorithm will have to allow for a rectangular region that can be divided in two either horizontally or vertically. If you imagine putting together the results of a horizontal and a vertical division, you will see that the best rectangular region to sum over may lie entirely within one of the pieces, may overlap two of them or may overlap all four of them. To allow for all this efficiently, it turns out that vectors of information are needed. We'll leave it to you to work further if you're interested.

Section 8.1

8.1.1. This is exactly the situation in the text, *except* that there is now one additional question when one reaches a leaf.

8.1.2. There are only two rooted binary trees with 4 leaves. (These are just rooted—not planar.) They are  and . Every path to a leaf in the first tree has length two and so $\mathrm{AC}(T) = 2$. If the frequencies are $f_1, \ldots, f_4$ from left to right, the average cost for the second tree is $3f_1 + 3f_2 + 2f_3 + f_4$, which in our case is $1.9 < 2$.

8.1.3 (a) If T is not a full binary tree, there is some vertex v that has only one child, say s. Shrink the edge (v, s) so that v and s become one vertex and call the new tree T'. If there are k leaves in the subtree whose root is v, then $\mathrm{TC}(T') = \mathrm{TC}(T) - k$.

(b) We follow the hint. Let k be the number of leaves in the subtree rooted at v. Since T is a binary tree and v is not a root, $k \geq 2$ Let $d = h(v) - h(l_2)$ and note that $d = (h(l_1) - 1) - h(l_2) \geq 1$. The distance to the root of every vertex in the subtree rooted at v is decreased by d and the distance of l_2 to the root is increased by d. Thus TC is decreased by $kd - d > 0$.

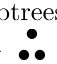
(c) By the discussion in the proof of the theorem, we know that the height of T must be at least m because a binary tree of height $m - 1$ or less has at most 2^{m-1}. Suppose T had height $M > m$.

By the previous part of this exercise, the leaves of T have heights M and, perhaps, $M-1$. Thus, every vertex of v with $h(v) < M-1$ has two children. It follows that T has 2^{M-1} vertices w with $h(w) = M-1$. If these were all leaves, T would have $2^{M-1} \geq 2^m$ leaves; however, at least one vertex u with $d(u) = M-1$ is not a leaf. Since it has two children, T has at least $2^m + 1$ leaves, a contradiction.

(d) By the previous two parts, all leaves of T have height at most m. If T' is principal subtree of T, its leaves have height at most $m-1$ in T'. Hence T' has at most 2^{m-1} leaves.

The argument hints at how to construct the desired tree: Construct T', a principal subtree of T, having all its leaves at height $m-1$ in T'. Construct a binary tree T'' having $n - 2^{m-1}$ such that $\mathrm{TC}(T'')$ is as small as possible. The principal subtrees of T will be T' and T''.

8.1.4. With one exception, the proof in the text can be used with k replacing 2. The exception occurs when we consider the principal subtrees of T. As in the text, we can dismiss the case of only one principal subtree. We must deal with d principal subtrees where $1 < d \leq k$. (When $k = 2$, it follows that $d = 2$ as in the text.) The function whose minimum is given in the exercise replaces $f(x)$ in the text.

8.1.5 (a) Suppose the answer is S_n. Clearly $S_0 = 1$ since the root is the only vertex. We need a recursion for S_n. One approach is to look at the two principal subtrees. Another is to look at what happens when we add a new "layer" by replacing each leaf with 

For the first approach, $S_n = 1 + 2S_{n-1}$, where each S_{n-1} is due to a principal subtree and the 1 is due to the root. The result follows by induction:

$$S_n \;=\; 1 + 2S_{n-1} \;=\; 1 + 2(2^n - 1) \;=\; 2^{n+1} - 1.$$

For the second approach, $S_n = S_{n-1} + 2^n$ and so $S_n = (2^n - 1) + 2^n = 2^{n+1} - 1$. By the way, if we have both recursions, we can avoid induction since we can solve the two equations

$$S_n \;=\; 1 + 2S_{n-1} \qquad \text{and} \qquad S_n \;=\; S_{n-1} + 2^n$$

to obtain the formula for S_n. Thus, by counting in two ways (the two recursions), we don't need to be given the formula ahead of time since we can solve for it.

(b) Let the value be $\mathrm{TC}^*(n)$. Again, we use induction and there are two approaches to obtaining a recursion. Clearly $\mathrm{TC}^*(1) = 0$, which agrees with the formula.

The first approach to a recursion: Since the principal subtrees of T each store S_{n-1} keys and since the path lengths all increase by 1 when we adjoin the principal subtrees to a new root, $\mathrm{TC}^*(n) = 2(S_{n-1} + \mathrm{TC}^*(n-1))$. Thus

$$\mathrm{TC}^*(n) \;=\; 2(2^n - 1 + (n-2)2^n + 2) \;=\; 2((n-1)2^n + 1) \;=\; (n-1)2^{n+1} + 2.$$

For the second approach, $\mathrm{TC}^*(n) = \mathrm{TC}^*(n-1) + n2^n$. Again, we can prove the formula for $\mathrm{TC}^*(n)$ by induction or, as in (a), we can solve the two recursions directly.

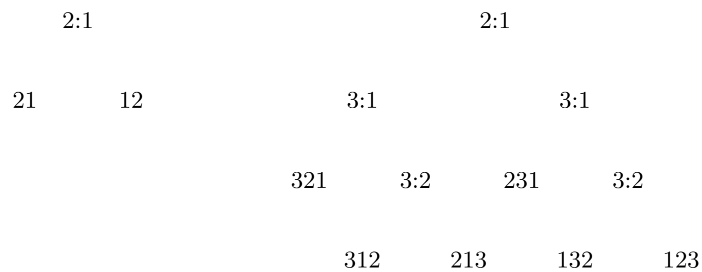


Figure S.8.1 The decision trees for binary insertion sorts. Go to the left at vertex $i:j$ if $u_i < s_j$ and to the right otherwise. (You may have done the reverse and gotten the mirror images. That's fine.)

Section 8.2

8.2.1. Here are the first few and the last.

1. Start the sorted list with 9.
2. Compare 15 with 9 and decide to place it to the right giving 9, 15.
3. Compare 6 with 9 to get 6, 9, 15.
4. Compare 12 with 9 and then with 15 to get 6, 9, 12, 15.
5. Compare 3 with 9 and then with 6 to get 3, 6, 9, 12, 15.

   . . . . . .

16. We now have the sorted list 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16. Compare 8 with 7, with 12 with 10 and then with 9 to decide where it belongs.

8.2.2. At each comparison, we roughly halve the number of places in which u_t could belong. This halving process must continue until just one position is left. After c comparisons, we have about $(\cdots((t/2)/2)\cdots/2) = t/2^c$ positions left. Thus $t/2^c$ is about 1 and so c is about $\log_2 t$. (This will be exact if t is a power of 2.) The total number of comparisons needed is about $\sum_{t=1}^{n} log_2 t$, which is about $\int_1^n \log_2 t \, dt$, which is about $nlog_2 n$.

8.2.3. See Figure S.8.1. To illustrate, suppose the original list is 3,1,2. Thus $u_1 = 3$, $u_2 = 1$ and $u_3 = 2$.

- We start by putting u_1 in the sorted list, so we have $s_1 = 3$.
- Now u_2 must be inserted into the list s_1. We compare u_2 with s_1, the 2:1 entry. Since $s_1 = 3 > 1 = u_2$, we go to the left and our sorted list is $1, 3$ so now $s_1 = 1$ and $s_2 = 3$.
- Now u_3 must be inserted into the list s_1, s_2. Since we are at 3:1, we compare $u_3 = 2$ with $s_1 = 1$ and go to the right. At this point we know that u_3 must be inserted into the list s_2. We compare $u_3 = 2$ with $s_2 = 3$ at 3:2 and go to the left.

8.2.4. First we sort on the units digit with 4 buckets:

$$1: 41,\ 21 \qquad 2: \textit{empty} \qquad 3: 33 \qquad 4: 14,\ 24.$$

Collect these in a list, preserving the order: 41, 21, 33, 14, 24. Now sort on the tens digit, preserving order:

$$1: 14 \qquad 2: 21,\ 24 \qquad 3: 33 \qquad 4: 41.$$

Collect these in a list, preserving the order: 14, 21, 24, 33, 31.

8.2.5 (a) Suppose that the alphabet has L letters and let the ith letter (in order) be a_i. Let u_j be a word with exactly k letters. The following algorithm sorts $u_1, \ldots, u_n$ and returns the result as $x_1, \ldots, x_n$.

```
BUCKET(u_1,...,u_n)
     Copy u_1,...,u_n to x_1,...,x_n.
     /* t is the position in the word. */
     For t = k to 1
     /* Make the buckets. */
          Create L empty ordered lists.
          For j = 1 to n
               If the tth letter of x_j is a_i,
                    then place x_j at the end of the ith list.
               End for
          Copy the ordered lists to x_1,...,x_n, starting with
               the first item in the first list and ending with the
               last item in the Lth list.
          End for
End
```

(b) Extend all words to k letters by introducing a new letter called "blank" which precedes all other letters alphabetically. Apply the algorithm in (a).

8.2.6. We only do the first algorithm since the second is built from it in a simple fashion. Let $\mathcal{A}(p)$ be the assertion that after p steps the words $x_1, \ldots, x_n$ are in order if we ignore the first $k-p$ letters of each word. To prove $\mathcal{A}(0)$, note that we are ignoring all the letters and so any order is fine. We now prove $\mathcal{A}(p)$ for $k \geq p > 0$ using $\mathcal{A}(p-1)$. Look at the situation after arranging the x_i's in the L ordered lists. If two words have different pth letters, they appear in different lists, the one with the alphabetically earlier letter appearing in the earlier list. Therefore, they will be in proper order when the L lists are copied into the x_i's. If two words have the same pth letter, they appear in the same list and their order in that list is the same as it was before they were placed there. Since that order was correct for their last $p-1$ letters by $\mathcal{A}(p-1)$, they will be in the proper order when copied into the x_i's.

8.2.7. First divide the list into two equally long tapes, say

A: 9, 15, 6, 12, 3, 7, 11 5   B: 14, 1, 10, 4, 2, 13, 16, 8.

Think of each tape as containing a series of 1 long (sorted) lists. (The commas don't appear on the tapes, they're just there to help you see where the lists end.) Merge the first half of each tape, list by list, to tape C and the last halves to D. This gives us the following tapes containing a series of 2 long sorted lists:

C: 9 14, 1 15, 6 10, 4 12   D: 2 3, 7 13, 11 16, 5 8.

Now we merge these 2 long lists to get 4 long lists, writing the results on A and B:

A: 2 3 9 14, 1 7 13 15   B: 6 10 11 16, 4 5 8 12.

Merging back to C and D gives

C: 2 3 6 9 10 11 14 16   D: 1 4 5 7 8 12 13 15.

These are merged to produce one 16 long list on A and nothing on B.

8.2.8. Read the items in k at a time, sort the k items and write them out onto tapes A and B, the first k onto A, the next k onto B, the next k onto A, etc. Now do a merge sort using A and B, starting with k long lists on each tape instead of 1 long lists. Note that if $k = 2^m$, then we have avoided m iterations of the merge sort.

8.2.9. A split requires $n-1$ comparisons since the chosen item must be compared with every other item in the list In the worst case, we may split an n long list into one of length 1 and another of length $n-1$. We then apply Quicksort to the list of length $n-1$. If $W(n)$ comparisons are needed, then $W(1)=0$ and $W(n) = n-1+w(n-1)$ for $n>1$. Thus $W(n) = \sum_{k=1}^{n-1} k = n(n-1)/2$.

Suppose that $n = 2^k$ and the lists are split evenly. Let $E(k)$ be the number of comparisons. Since Quicksort is applied to two lists of length $n/2$ after splitting, $E(k) = n-1+2E(k-1)$ for $k>0$ and $E(0)=0$. A little computation gives us $E(1)=1$, $E(2)=5$, $E(3)=17$, $E(4)=49$ and $E(5)=129$. From the statement of the problem we expect $E(k)$ to be near $k2^k$, which has the values 0, 2, 8, 24 and 64. Comparing these sequences we discover that $E(k) = 2(k-1)2^{k-1}+1$ for $k<6$. This is easily proved by induction

8.2.10 (a) The number of comparisons is actually $(n-1)+Q(rn)+Q((1-r)n)$ by the same argument used in the previous exercise.

(b) Suppose the claim is true for rn and $(1-r)n$, which are both less than n. By the previous part, $Q(n)$ is about

$$\begin{aligned} & n + a\,rn\ln(rn) + a\,(1-r)n\ln((1-r)n) \\ &= \; n + arn\ln n + arn\ln r + a(1-r)n\ln n + a(1-r)n\ln(1-r) \\ & an\ln n + n[1 + ar\ln r + a(1-r)\ln(1-r)]. \end{aligned}$$

If this is to equal $an\ln n$, the factor $[\cdots]$ must be 0. (This is not a rigorous inductive proof because we've loosely thrown around the word "about.")

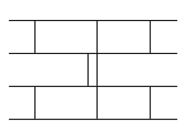
(c) By the previous problem, the answer is about $n\log_2 n$. Our formula for a is $1 + a\ln(1/2)$ and so $a = 1/\ln 2$. Since $(n\ln n)/(\ln 2) = n\log_2 n$, we are done.

(d) The equation for a is $1 - a(\ln 3 + 2\ln(3/2))/3 = 0$ and so $a = 3/\ln(27/4)$. If we are using natural logarithms, $a = 1.57$.

Section 8.3

8.3.1. Since there are only 3 things, you cannot compare more than one pair of things at any time. By the Theorem 8.1, we need at least $\log_2(3!)$ comparisons; i.e., at least three. A network with three comparisons that sorts is given in Figure 8.2.

8.3.2. See the previous solution. We need at least $\log_2(4!)$ comparisons (i.e., we need five or more). We can compare two pairs at the same time, so the fastest network must take at least three time units. Here is a network that does it.



To prove that it sorts you could look at all $4! = 24$ possible permutations of the inputs. It's easier to use the Zero-One Principle.

8.3.3. As argued in the previous two solutions, we will need at least seven comparisons and we can do at least two per time. This means it will take at least four time units. It has been shown (but not in this text!) that at least five time units are required. A brick wall sort works.

8.3.4. If you have an upper or lower bound on the values to be sorted, you can "pad out" the list with one of these values to obtain n items. The added items will be at one end. What if you don't know such a bound? Now it depends on your situation a bit more. By one pass through the items one can find an upper (or lower) bound. If this is not desirable, you can pad the list with copies of anything and then pass the sorted items through a hardware or software device to remove the pads.

8.3.5. One possibility is the type of network shown in Figure 8.3. For n inputs, this has

$$1 + 2 + \cdots + (n-1) \;=\; \frac{n(n-1)}{2}$$

comparators. It was noted in the text that a brick wall for n items must have length n. If n is even there are $(n/2)(n-1)$ comparators and if n is odd there are $n((n-1)/2)$ comparators. Thus this is the same as Figure 8.3. We don't know if it can be done with less.

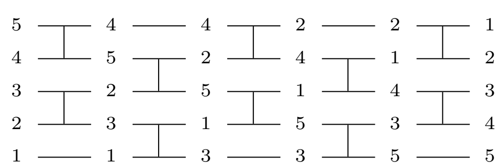
8.3.6 (a) If the network is used k times, it is the same as $2k$ time units of the brick wall. To see this, write down a new copy of the network each time you are supposed to feed it back in. For sorting to have taken place for all inputs, we must have $2k \geq n$. It follows that $f(n) = n$ for n even and $f(n) = n+1$ for n odd.

(b) This problem was discussed in the first exercise for a general network. There is a new twist here though. Suppose we wish to sort m items. Feed them in as the top m inputs and pad out the last $n-m$ inputs with an upper bound. The network will be done after being run $f(m)$ times. Another twist: forget the padding and disable the appropriate lower comparators. This works because the network for the first m lines is also a brick wall.

(c) If you write down what the comparators are, you will see that this is the brick wall again in disguise; however, the wires are now "rotating" around. We need to know when they've gotten back to their starting positions. That happens after n shifts. Thus we have sorted output after n time units provided we read what would be fed in as new input next time. If we read the output, $n+1$ time units are required.

8.3.7. By the Adjacent Comparisons Theorem, we need only check the sequence $n, \ldots, 2, 1$. Using the argument that proves the Zero-One Principle, it follows that this sequence is sorted if and only if all sequences that consist of a string of ones followed by a string of zeroes are sorted.

8.3.8 (a)



If you start at the leftmost 5, you can follow vertical and horizontal lines so that you pass just the fives and end at the rightmost 5. Imagine that the network is made of bars and standing upright as it is pictured. Now remove all the bars along which you have travelled and gently lower the upper portion so that it rests on the lower one at the same time sliding it to the left one time unit (or comparator). The result is a network that is sorting the sequence $4, 3, 2, 1$ by adjacent comparisons; in fact, the network is a brick wall. This will be clearer if you copy the network to a piece of paper, cut it along the path and slide the pieces together.

8.3.9. It is evident that the idea in the solution to (a) of the previous exercise works for any n. This can be used as the basis of an inductive proof.

An alternative proof can be given using sequences that consist of ones followed by zeroes. (See Exercise 8.3.7.) Note that when the lowest 1 starts moving down through the comparators, it moves down one line each time unit until it reaches the bottom. The 1 immediately above it starts the same process one time unit later. The 1 immediately above this one starts one more time unit later, and so forth. If there are j ones and the lowest 1 reaches the bottom after an exchange at time t, then the next 1 reaches proper position after an exchange at time $t+1$. Continuing in this way, all ones are in proper position after the exchanges at time $t+j-1$. Suppose the jth 1 (i.e., lowest 1) starts moving by an exchange at time i. Since it reaches position after $n-j$ exchanges, $t = i+(n-j)-1$. Thus all ones are in position after the exchanges at time $(i+(n-j)-1)+j-1 = n+i-2$. The jth 1 starts moving when it is compared with the line below it. This happens at time 1 or 2. Thus $n+i-2 \le n$.

8.3.11. Use induction on n. For $n = 2^1$, it works. Suppose that it works for all powers of 2 less than 2^t. We use the variation of the Zero-One Principle mentioned in the text. Suppose that the first half of the x_i's contains α zeroes and the second half contains β zeroes. `BMERGE` calls `BMERGE2` with $k = j = 2^{t-1}$ By the induction assumption, `BMERGE2` rearranges the "odd" sequence $x_1, x_3, \ldots, x_{2^t-1}$ in order and the "even" sequence $x_2, x_4, \ldots, x_{2^t}$ in order. The number of zeroes in the odd seqence minus the number of zeroes in the even sequence in 0, 1 or 2; depending on how many of α and β are odd. When the difference is 0 or 1, the result of `BMERGE2` is sorted. Otherwise, the last zero in the odd sequence, $x_{\alpha+\beta+1}$, is after the first one in the even sequence, $x_{\alpha+\beta}$, and all other x_i's are in order. The comparator in `BMERGE` with $i = (\alpha+\beta)/2$ fixes this.

8.3.12. This is essentially the same as when n is a power of 2; however, one must be a bit careful with the indices k and j.

8.3.13 (a) Since a one long list is sorted, nothing is done and so $S(0) = 0$. The two recursive calls of `BSORT` can be implemented by a network in which they run during the same time interval. This can then be followed by the `BMERGE` and so $S(N) \le S(N-1) + M(N)$.

(b) As for $S(0) = 0$, $M(0) = 0$ is trivial. Since all the comparators mentioned in `BMERGE` and be run in parallel at the same time, $M(N) \le M(N-1) + 1$.

(c) From the previous part, it easily follows by induction on N that $M(N) \le N$. Thus $S(N) \le S(N-1) + N$ and the desired result follows by induction on N.

(d) If $2^{N-1} < n \le 2^N$, then the above ideas show that $S(n) \le N(N+1)/2$. Thus

$$S(n) < \frac{1}{2}(1+\log_2 n)(2+\log_2 n).$$

Section 9.1

9.1.1. We do `PREV`(T).

```
PREV(T)
     Let r be the root of T
     Let T_1,...,T_k be the principal subtrees of T
     Output r
     For i = 1,...,k    Prev(T_i)
End
```

9.1.2. In the theorem, Step 4 corresponds to a return from a recursive call and Step 5 corresponds to a recursive call.

9.1.3. We give pseudocode for vertex visitation.

```
BFV(T)
     Initialize queue
     INQUEUE(T)
     While queue not empty
          S = OUTQUEUE( )
          Let r be the root of S
          Let S_1,...,S_k be the principal subtrees of S
          Output r
          For i = 1,...,k  INQUEUE(S_i)
     End while
End
```

9.1.4. Here is the pseudocode.

```
PREV(T)
     Initialize stack
     PUSH(T)
     While stack not empty
          S = POP( )
          Let r be the root of S
          Let S_1,...,S_k be the principal subtrees of S
          Output r
          For i = k,...,1  PUSH(S_i)
     End while
End
```

9.1.5. The proof can be done by induction on the size of the tree by showing that the comments in the algorithm are correct. To do this, we need to notice a couple of things.

- By removing r from G before constructing S, we guarantee that S will not contain r. Thus it will contain precisely the vertices that are reachable on a path from r, starting with the edge $\{r, s\}$.
- Because we remove the root vertex of the tree from G and do this recursively, whenever a tree is ready to return, all its vertices have been removed from G. As a result, none of the vertices in S are left in G when we construct R.

9.1.6 (a) There is only one tree with vertex set V when $|V| = 1$.

(b) After the root v of T, the next vertex visited in a depth-first traversal is the root of T_1.

(c) The root of T_1 will be listed in POSTV after all the other vertices of T_1 have been visited. The vertices listed previous to this will be exactly the other vertices of T_1 since they will not be visited again but all other vertices of T will be visited again.

(d) From the previous part, T_1 has t vertices. After listing the root v of T, PREV lists the t vertices of T_1 and then list other vertices of T.

(e) U is simply T with T_1 removed. Thus we can obtain the traversal sequences for U by traversing T and "forgetting" to list any vertices of T_1 that we encounter.

(f) Since T_1 and U have fewer vertices than T, they can both be reconstructed by induction. To get T, simply adjoin T_1 to U as a new leftmost child of v.

9.1.8. Here is the pseudocode.

```
PREVandPOSTV(a_1,...,a_n; z_1,...,z_n)
     If n = 1
          Return the single vertex tree a_1
     End if
     Define t by z_t = a_2
     T_1 =PREVandPOSTV(a_2,...,a_{t+1}; z_1,...,z_t)
     U =PREVandPOSTV(a_1,a_{t+2},...,a_n; z_{t+1},...z_n)
     T is the tree obtained by
           adding T_1 to U as a new leftmost child of a_1
     Return T
End
```

9.1.9 (a) $D(T)$ is $+1, D(T_1), -1, +1, D(T_2), -1, \ldots, +1, D(T_m), -1$.

(b) Each edge is traversed twice, proving the sum. The rest can be proved by induction on the number of vertices using the formula in (a). Actually, one can show more: The sum up to k is the length of the path from the root to the vertex that is reached after k steps.

(c) The "if" part follows from (b). The "only if" part can be done by showing that there is a unique way to construct a tree associated with such a sequence. This can be done recursively if we use the observation that the sum up to k is 0 if and only if we have returned to the root after k steps: Let k be the first index for which the sum is 0. We must have $s_1 = +1$, $s_k = -1$ and the subsequences $s_2, \ldots, s_{k-1}$ and $s_{k+1}, \ldots, s_n$ are associated with unique RP-trees. There is just one way to piece these trees together.

Section 9.2

9.2.1.

```
  (a)     +       (b)     +       (c)     +            (d)     /            (e)     +

        +   5           +   5           1   +              +       −             −       *

      +   4           +       +             2   +        X   *   X   Y         *   3   X   +

    +   3           1   2   3   4               3   +        5   Y           X   Y       X   1

  1   2                                             4   5
```

9.2.2. After "`End if`" insert

```
     If exp = −exp1
     Return the RP-tree with root ‘‘−’’
          and son INTERPRET(exp1).
```

9.2.3. We use `value(···)` to indicate the value of a variable or constant.

(a) The first method:

```
EVALUATE(exp)
     If (exp has no op)   Return value(exp).
     If (exp = −exp1)   Return −value(exp1).
     Let exp = (exp1 op ; exp2).
     Return EVALUATE(exp1) op EVALUATE(exp2).
End
```

(b) The second method:

```
EVALUATE(T)
     Let r be the root of T.
     Let k be the number of principal subtrees of T
          and let T_i be the ith of them.
     If (k = 0), Return value(r).
     For i = 1,...,k    Let v_i = EVALUATE(T_i).
     /* If k = 1, r should be unary minus. */
     If k = 1, Return r v_1.
     If k = 2, Return v_1 r v_2.
End
```

9.2.5. We will indicate what needs to be added. Other solutions are possible.

(a) *exp* $\rightarrow$ $-$ *term*

(b) *term* $\rightarrow$ *power* and *power* $\rightarrow$ *factor* | *factor* ∗∗ *power*

(c) Let *subst* be the start symbol now and add *subst* $\rightarrow$ *exp* | **id :=** *exp*

(d) This is a bit trickier because the **:=** must reach as far to the right as possible. In particular, you cannot replace the last three items in the following list with just *factor* $\rightarrow$ *subst*. Let *start* be the start symbol.

$$
\begin{aligned}
\textit{start} &\rightarrow \textit{exp} \mid \textit{subst}\\
\textit{subst} &\rightarrow \textbf{id :=}\ \textit{exp} \mid \textbf{id :=}\ \textit{subst}\\
\textit{exp} &\rightarrow \textit{exp}{+}\textit{subst} \mid \textit{exp}{-}\textit{subst}\\
\textit{term} &\rightarrow \textit{term}{*}\textit{subst} \mid \textit{term}/\textit{subst}\\
\textit{factor} &\rightarrow (\ \textit{subst}\ )
\end{aligned}
$$

9.2.6 (a) It consists of all strings of the form $\alpha_1 \pm \alpha_2 \cdots \pm \alpha_n$ where $n \geq 1$, α_i is **x** or **y** and $\pm$ is either "+" or "−", except that **y** cannot follow "−".

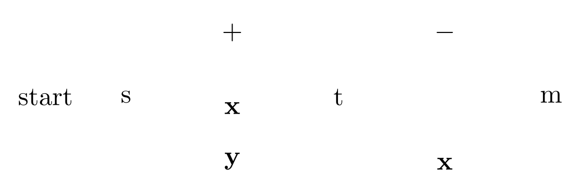
(b) G' is G. The grammar for G'' is

$$
\begin{aligned}
s &\rightarrow \mathbf{x}\,t \mid \mathbf{y}\,t\\
t &\rightarrow +(t, +\mathbf{x}t, 2) \mid -(t, -\mathbf{x}t, 2) \mid +(t, +\mathbf{y}t, 2) \mid \text{the-empty-string}\\
(t, +\mathbf{x}t, 2) &\rightarrow \mathbf{x}t\\
(t, -\mathbf{x}t, 2) &\rightarrow \mathbf{x}t\\
(t, +\mathbf{y}t, 2) &\rightarrow \mathbf{y}t
\end{aligned}
$$

The machine has four states corresponding to the left sides of the four productions just given. The start state is s and the accepting state is t. You should be able to draw it.

(c) The machine is deterministic.

(d) Yes. One can construct a three state machine with start state s and accepting state t as shown here.



Section 9.3

9.3.1. The construction starts with •. The first iteration gives • and all trees that have children produced in the starting step. Thus we get

In the next iteration, we obtain the following new trees with at most 4 vertices.

In the next step, the only new tree is a 4-vertex tree consisting of a path from the root to a single leaf. After this, no new trees with less than 5 vertices are obtained.

9.3.2. The construction gives the trees in Figure 9.5, then the next step produces 4 new trees. After that, no new trees with less than 5 leaves are obtained.

9.3.3. For $k \leq 7$, the values are in the text. $b_8 = 429$, $b_9 = 1430$ and $b_{10} = 4862$.

9.3.4. Here are the calculations:

$$(b_1b_4 + b_2b_3) + 0 \times b_2 + 0 \;=\; 7$$
$$(b_1b_5 + b_2b_4) + (b_1b_2 + 0 \times b_1 + 0) \times b_3 + 0 \;=\; 21$$
$$(b_1b_5 + b_2b_4) + 0 \times b_2 + 0 \;=\; 19.$$

9.3.5. We'll use $n\colon r$ to mean a tree with n leaves and rank r and $(n_1\colon r_1,\ n_2\colon r_2)$ to mean a tree with left son $n_1\colon r_1$ and right son $n_2\colon r_2$. We use formula (9.4) and the greedy approach: First make $|T_1|$ as large as possible, then make $\text{RANK}(T_1)$ as large as possible. Here are the calculations. You should be able to construct the trees easily from the results as long as you remember (a) that $n\colon 0$ describes a tree in which all the left sons are leaves (since that is the leftmost tree in the list of trees) and (b) that since there is only one tree with 1 leaf and only one with 2 leaves, they each have rank 0.

$$\begin{aligned}
8\colon 100 &= (1\colon 0,\ 7\colon 100) && \text{since } b_1b_7 = 132 > 100, \text{ we have } |T_1| = 1 \text{ and } 100 = 0b_7 + 100\\
7\colon 100 &= (6\colon 10,\ 1\colon 0) && \text{since } b_1b_6 + \cdots + b_5b_2 = 90 \text{ and } 10 = 10b_1 + 0\\
6\colon 10 &= (1\colon 0,\ 5\colon 10) && \text{since } b_1b_5 = 14 > 10 \text{ and } 10 = 0b_5 + 10\\
5\colon 10 &= (4\colon 1,\ 1\colon 0) && \text{since } b_1b_4 + \cdots + b_3b_2 = 9 \text{ and } 1 = 1b_1 + 0\\
4\colon 1 &= (1\colon 0,\ 3\colon 1)\\
3\colon 1 &= (2\colon 0,\ 1\colon 0)
\end{aligned}$$

$$\begin{aligned}
8\colon 200 &= (3\colon 1,\ 5\colon 12) && \text{since } b_1b_7 + b_2b_6 = 174 \text{ and } 26 = 1b_5 + 12\\
5\colon 12 &= (4\colon 3,\ 1\colon 0) && \text{since } b_1b_4 + b_2b_3 + b_3b_2 = 9\\
4\colon 3 &= (3\colon 0,\ 1\colon 0) && \text{since } b_1b_3 + b_2b_2 = 3
\end{aligned}$$

$$\begin{aligned}
8\colon 300 &= (7\colon 3,\ 1\colon 0)\\
n\colon 3 &= (1\colon 0,\ n-1\colon 3) && \text{for } n \geq 5 \text{ since } b_1b_{n-1} > 3\\
4\colon 3 &= (3\colon 0,\ 1\colon 0)
\end{aligned}$$

$$8\colon 400 = (7\colon 103,\ 1\colon 0) \quad 7\colon 103 = (6\colon 13,\ 1\colon 0) \quad 6\colon 13 = (1\colon 0,\ 5\colon 13)$$
$$5\colon 13 = (4\colon 4,\ 1\colon 0) \quad 4\colon 4 = (3\colon 1,\ 1\colon 0)$$

9.3.6. We will induct on n. The result is true for $n = 1$.

Suppose that T is a tree with $n > 1$ leaves. The root of T has a left son with, say $k > 0$ leaves. Thus the right son has $n - k > 0$ leaves. It follows that $k < n$ and $n - k < n$ and so by induction, the left and right sons have a total of $(k-1) + (n-k-1) = n-2$ other vertices. Counting the root, we see that T has $(n-2) + 1 = n-1$ other vertices.

9.3.7 (b) In the notation introduced in Exercise 9.3.5, with $n = 2m+1$ and $k = b_n/2$, we claim that $\mathcal{M}_n = n{:}k = (m+1{:}0,\ m{:}0)$. To prove this, note that the rank of this tree is $b_1b_{2m} + \cdots + b_mb_{m+1}$ and that

$$b_n \;=\; b_1b_{2m} + \cdots + b_{2m}b_1 \;=\; 2(b_1b_{2m} + \cdots + b_mb_{m+1}).$$

(c) If $n = 2m$, then $b_n = 2(b_1b_{2m-1} + \cdots + b_{m-1}b_{m+1}) + b_m^2$, which is divisible by 2 if and only if b_m is. Thus there is no such tree unless b_m is even. In this case you should be able to show that $\mathcal{M}_{2m} = (m{:}b_m/2,\ m{:}0)$.

9.3.8. Let the function in one line form be $(f_1,\ldots,f_k)$ and denote the rank by $\mathrm{RANK}(f_1,\ldots,f_k)$. Then $\mathrm{RANK}(n) = n-1$ and, for $k>1$ arguments, $\mathrm{RANK}(a,b,\ldots,z) = \binom{a-1}{k} + \mathrm{RANK}(b,\ldots,z)$.

9.3.10. If f is a permutation of $\underline{k}$, define f^*, a permutation of $\underline{k-1}$, by

$$f^*(i) \;=\; f(i+1) - H(f(i+1) - f(1)), \quad \text{where} \quad H(x) \;=\; \begin{cases} 0, & \text{if } x<0, \\ 1, & \text{otherwise.} \end{cases}$$

In effect, f^* is f with the first value removed and the rest pushed down so that the range becomes $\underline{k-1}$. Let p be a permutation of $\underline{n}$. Define

$$\mathrm{Lex}(p,n) \;=\; \begin{cases} 0, & \text{if } n=1, \\ (n-1)!\,\big(p(1)-1\big) + \mathrm{Lex}(p^*, n-1), & \text{if } n>1. \end{cases}$$

We leave it to you to convince yourself that this function does compute the lex order rank of p.

9.3.11 (a) An x_i belongs in a parenthesis pair that has nothing inside it. Number the empty pairs from left to right and insert x_i into the i^{th} pair.

(b) This is just a translation of what has been said. If you are confused, remember that $\mathtt{B}(n)$ should is to be thought of as all possible parentheses patterns for $x_1,\ldots,x_n$.

(c) This simply involves the replacement described in (b): Make $\bullet$ correspond to ( ) and make the tree with sons T_1 and T_2 correspond to (P_1P_2), where P_i is the parentheses pattern corresponding to T_i.

9.3.12 (a) We prove by induction on the number of vertices that f maps a tree with n vertices to one with n leaves. This is certainly true for the single vertex tree, $[\,] = \bullet$. Let $L(T)$ and $V(T)$ be the number of leaves and vertices of a tree. We have

$$\begin{aligned} L\Big(f\big([T_1,\ldots,T_k]\big)\Big) &= L\Big(f(T_1)\Big) + L\Big(f\big([T_2,\ldots,T_k]\big)\Big) && \text{by (9.5);} \\ &= V(T_1) + V\big([T_2,\ldots,T_k]\big) && \text{by induction;} \\ &= V\big([T_1,\ldots,T_k]\big). \end{aligned}$$

An easy way to see that f is a bijection is to exhibit its inverse. Let $g(\bullet) = \bullet$ and $g\big([B_1,B_2]\big) = \big[g(B_1), S_1,\ldots,S_j\big]$, where $[B_1,B_2]$ is an unlabeled full binary RP-tree and $g(B_2) = [S_1,\ldots,S_j]$. One can prove by induction that $g(f(T)) = T$ and $f(G(B)) = B$. We do the latter using the above definition of the S_i's.

$$\begin{aligned} f\big(g(B)\big) &= f\Big(\big[g(B_1), S_1,\ldots,S_j\big]\Big) && \text{by defn. of } g; \\ &= \Big[f\big(g(B_1)\big), f\big([S_1,\ldots,S_j]\big)\Big] && \text{by defn. of } f; \\ &= \Big[f\big(g(B_1)\big), f\big(g(B_2)\big)\Big] && \text{by defn. of } S_i\text{'s;} \\ &= [B_1,B_2] \;=\; B && \text{by induction.} \end{aligned}$$

(b) We have introduced an operation called `JOIN` for two full binary RP-trees. Redefine it for any two RP-trees S and $T = [T_1,\ldots,T_k]$ to be $[S,T_1,\ldots,T_k]$.

9.3.13 (a) The leaves in an RP-tree are distinguishable because the tree is ordered. Thus, each marking of the n leaves leads to a different situation. The same comments applie to vertices and there are $2n-1$ vertices by Exercise 9.3.6.

(b) Mark the single vertex that arises in this way to obtain an element of $\mathcal{V}_n$ Interchanging x and the tree rooted at b gives a different element of $\mathcal{L}_{n+1}$ that gives rise to the same element of $\mathcal{V}_n$.

Conversely, given any element of $\mathcal{V}_n$, the marked vertex should be split into two, f and b with b a son of f. Introduce another son x of f which is a marked leaf. There are two possibilities–make f a left son or a right son.

(c) By (a), $|\mathcal{L}_n| = nb_n$ and $|\mathcal{V}_n| = (2n-1)b_n$. By (b), $|\mathcal{L}_{n+1}| = 2|\mathcal{V}_n|$.

(d) By the recursion,

$$b_n = \frac{2(2n-3)}{n} b_{n-1} = \frac{2(2n-3)}{n}\frac{2(2n-5)}{n-1} b_{n-2} = \cdots = \frac{2^{n-1}(2n-3)(2n-5)\cdots 1}{n(n-1)\cdots 2} b_1.$$

Using $b_1 = 1$, we have a simple formula; however, it can be written more compactly:

$$\begin{aligned} b_n &= \frac{2^{n-1}(2n-3)(2n-5)\cdots 1}{n!} = \frac{2^{n-1}(n-1)!\,(2n-3)(2n-5)\cdots 1}{(n-1)!\,n!} \\ &= \frac{(2n-2)!}{(n-1)!\,n!} = \frac{1}{n}\binom{2n-2}{n-1}. \end{aligned}$$

Section 10.1

10.1.1 (d) Note that $r = 1 + x + x^2 + x^3 + \cdots$. Whenever you add, subtract or multiply power series and look for the coefficient of some power of x, say x^n, only those powers of x that do not exceed n in the original series matter. Each of p, q and r begin $1 + x + x^2 + x^3$.

10.1.2. We can neglect all terms above x^2 at each step of our calculations. We'll use $\approx$ to indicate that we've neglected such terms.

(a) $(2+x+x^2)(1+2x+x^2)(1+x+2x^2) \approx (2+5x+5x^2)(1+x+2x^2)$ and so the answer is 14.

(b) $(1+2x+x^2)^2 \approx 1+4x+6x^2$, $(1+x+2x^2)^3 \approx 1+3x+9x^2$, $(1+4x+6x^2)(1+3x+9x^2) \approx 1+7x+27x^2$, and so the answer is 62.

(c) This is the same as the coefficient of x in $(1+x)^{43}(2-x)^5$, so we need keep only constant and linear terms. Thus $(1+x)^{43} \approx 1+43x$ and $(2-x)^5 \approx 2^5 - 5\times 2^4 x = 32 - 80x$. The answer is $32 \times 43 - 80 = 1296$.

10.1.3. We have

$$(x^2+x^3+x^4+x^5+x^6)^8 = x^{16}(1+x+x^2+x^3+x^4)^8 = x^{16}\left(\frac{1-x^5}{1-x}\right)^8.$$

The coefficient of x^{21} in this is the coefficient of x^5 in the eighth power on the right hand side. Since $(1-x^5)^8 = 1-8x^5+\cdots$, this is simply the coefficient of x^5 in $(1-x)^{-8}$ minus 8 times the coefficient of x^0 (the constant term) in $(1-x)^{-8}$. Thus our answer is

$$(-1)^5\binom{-8}{5} - 8 = \binom{12}{5} - 8 = 784.$$

10.1.4 (a) Let $f(z) = (1+z)^r$. It is easy to show that

$$f^{(k)}(z) = r(r-1)\cdots(r-k+1)(1+z)^{r-k}.$$

Thus $f^{(k)}(0)/k! = \binom{r}{k}$. Taylor's Theorem completes the derivation as long as we ignore convergence.

To prove convergence, we need to verify that the remainder term in Taylor's formula goes to 0 as $k \to \infty$. We'll do this for $C = 1/3$. Taylor's Theorem with remainder states that

$$f(x) = \sum_{n=0}^{k-1} \frac{f^{(n)}(0)x^n}{n!} + R_k(x) \quad \text{where} \quad |R_k(x)| \le \frac{|x|^k}{k!}M$$

and $M = \max_{|t|\le|x|} |f^{(k)}(t)|$. From the above calculations, $f^{(k)}(t)/k!, ; = \binom{r}{k}(1+t)^{r-k}$.

We first bound the binomial coefficient. Since we are interested in convergence, *we can assume that k is as large as we wish.* If $r \ge 0$, then $|r-k| \le k$ when k is large enough and so

$$\left|\binom{r}{k}\right| = \left|\frac{r}{1}\right| \left|\frac{r-1}{2}\right| \cdots \left|\frac{r-k}{k}\right|$$

is bounded because all but the first few factors are 1 or less. On the other hand, if $r < 0$ let m be an integer greater than $-r$ and note that

$$\left|\binom{r}{k}\right| < \frac{1}{(m-1)!} \frac{-r}{m} \frac{-r+1}{m+1} \cdots \frac{-r+k-m}{k} (k-r)^{m-1}.$$

Since all but the last factor is at most 1, it follows that this expression is bounded by a constant (depending on r) times k^{m-1}. Putting all this together, there are constants A and B depending on r so that $\left|\binom{r}{k}\right| < Ak^B$ when k is large.

Now suppose that $|t| \le |x| \le 1/3$. Then for large k, $k - r > 0$ and

$$|R_k(x)| \le Ak^B \frac{\max_x |x|^k}{\min_t (1+t)^{k-r}} \le Ak^B \frac{(1/3)^k}{(1-1/3)^{k-r}} = A(2/3)^r k^B (1/2)^k,$$

which goes to 0 as $k \to \infty$.

(b) In the previous result, set $r = -1$. Then $\binom{r}{k} = \binom{-1}{k} = (-1)^k$ and $z^k = (-1)^k z^k$. Thus $(1-z)^{-1} = \sum (-1)^k (-1)^k z^k = \sum z^k$. Multiply both sides by a.

(c) You may be familiar with this formula: it's the sum of a (finite) geometric series. There are various ways to do obtain the result. We'll give two.

First, from scratch. Let $S = \sum_{k=0}^n az^k$. Note that the k^{th} term of zS equals the $(k+1)^{\text{st}}$ term of S. Thus almost all terms cancel if we subtract zS from S. In fact, $S - zS = a - az^{n+1}$. Solve this for S.

Second, from (b). Note that with S as above and $i = k - (n+1)$,

$$\sum_{k\ge 0} az^k = S + \sum_{k\ge n+1} az^k = S + \sum_{i\ge 0} (az^{n+1})z^i.$$

By (b), the leftmost sum is $a/(1-z)$ and the rightmost sum is $(az^{n+1})/(1-z)$. Solve this for S.

(d) Use (a) with $(z, r) = (-ax, -2)$. Note that $\binom{-2}{k} = (-1)^k(k+1)$. Thus the coefficient of x^n is $(n+1)a^n$.

10.1.5. We'll do just the general k.

(a) We have $x^kA(x) = \sum_{m\ge0} a_m x^{m+k} = \sum_{n\ge k} a_{n-k}x^n$. Thus the coefficient of x^n is 0 for $n<k$ and a_{n-k} for $n \ge k$.

(b) We have

$$\left(\frac{d}{dx}\right)^k A(x) = \sum_{m=0}^{\infty} a_m \left(\frac{d}{dx}\right)^k x^m = \sum_{m=k}^{\infty} a_m(m)(m-1)\cdots(m-k+1)x^{m-k}.$$

Set $n = m-k$ to obtain the answer: $a_{n+k}(n+k)(n+k-1)\cdots(n+1) = a_{n+k}\frac{(n+k)!}{n!}$.

(c) Since $\left(x\frac{d}{dx}\right)A(x) = \sum_{m=0}^{\infty} ma_mx^m$, repeating the operation k times leads to $\sum_{m=0}^{\infty} m^ka_mx^m$. Thus the answer is n^ka_n.

10.1.6 (a) Let $b_n = 1$ for all $n \ge 0$ in the theorem. Then $C(x) = A(x)/(1-x)$.

(b) Since we are summing on k, not n, this may be a bit confusing. We apply (a) with the n in (a) replaced by k and $a_j = (-1)^j\binom{n}{j}$. Thus the generating function for the sum is $C(x) = A(x)/(1-x)$. By the Binomial Theorem (p. 17), $A(x) = (1-x)^n$. Thus $B(x) = (1-x)^{n-1}$ and so, by the Binomial Theorem again, the answer is $(-1)^k\binom{n-1}{k}$.

(c) You should be able to see that

$$d_n = \sum_{i=0}^{n} a_ie_{n-i} \quad \text{where} \quad e_m = \sum_{j=0}^{m} b_jc_{m-j}.$$

Thus $D(x) = A(x)E(x)$ and $E(x) = B(x)C(x)$, and so $D(x) = A(x)B(x)C(x)$.

10.1.7. This is simply the derivation of (10.4) with r used instead of 1/3. The generating function for the sum is $S(x) = 1/(1-r(1+x))$ and the coefficient of x^k is

$$\frac{\big(r/(1-r)\big)^k}{1-r} = \frac{\big(r/(1-r)\big)^{k+1}}{r}.$$

To verify convergence, let $a_n = \binom{n}{k}r^n$ and note that

$$\lim_{n\to\infty}\left|\frac{a_{n+1}}{a_n}\right| = \lim_{n\to\infty}\frac{(n+1)|r|}{n-k+1} = |r| < 1.$$

10.1.8. The main difficulty here is understanding what corresponds to what. In the convolution formula, an index k ranges from 0 to n in the summation and here i ranges from 0 to k. Thus (n,k) in the convolution formula corresponds to (k,i) here. The values of n and m here are simply constants. Let's choose the unused variable j for an index in hopes of avoiding some confusion. We have $a_j = \binom{m}{j}$ and $b_j = \binom{n}{j}$. By the Binomial Theorem, $A(x) = (1+x)^m$ and $B(x) = (1+x)^n$. Thus $C(x) = A(x)B(x) = (1+x)^{m+n}$. By the Binomial Theorem, $c_j = \binom{m+n}{j}$.

10.1.9. This is very similar to the Exercise 10.1.8 With $a_j = (-1)^j\binom{m}{j}$ and $b_j = \binom{m}{j}$, we can apply the convolution formula. The result is $C(x) = (1-x)^m(1+x)^m = (1-x^2)^m$. By the binomial theorem, $(1-x^2)^m = \sum(-1)^j\binom{m}{j}x^{2j}$. Thus, the sum we are to simplify is zero if k is odd and $(-1)^j\binom{m}{j}$ if $k = 2j$.

10.1.10. It's simply a matter of expanding generating functions and looking at coefficients, realizing that $(-1)^n$ is $+1$ or -1 according as n is even or odd. The answer for (b) is $(A(x)-A(-x))/2$. To answer (c), use (a) with $A(x) = (1+x)^n$. The result is $\frac{1}{2}\Big((1+x)^n + (1-x)^n\Big)$. When $x=1$ we obtain 2^{n-1}.

10.1.11. The essential fact that $\sum_{s=0}^{k-1}\omega^{rs}$ is k if r is multiple of k and 0 otherwise.

10.1.12. As a first method, we'll follow the example in the text. If $F(x,y) = \sum_{n,k}\binom{2n}{k}x^k y^n$, then the generating function for s_k is $S(x) = F(x,1/2)$. We have

$$F(x,y) \;=\; \sum_{n=0}^{\infty}(1+x)^{2n}y^n \;=\; \sum_{n=0}^{\infty}\Big((1+x)^2 y\Big)^n \;=\; \frac{1}{1-(1+x)^2 y}.$$

Hence $S(x) = 2/(1-2x-x^2)$. This can be done by partial fractions as discussed in the next section.

Now we do the problem using bisection of series Let $G(x,y) = \sum_{n,k}\binom{n}{k}x^k y^n = \Big(1-(1+x)y\Big)^{-1}$. We use bisection to extract the part with n even and then set $y = 1/\sqrt{2}$ to obtain $S(x)$:

$$\begin{aligned} S(x) &= \left.\frac{G(x,y)-G(x,-y)}{2}\right|_{y=1/\sqrt{2}} \\ &= \frac{\Big(1-(1+x)/\sqrt{2}\Big)^{-1} + \Big(1+(1+x)/\sqrt{2}\Big)^{-1}}{2} \\ &= \frac{\sqrt{2}}{\sqrt{2}-1}\,\frac{1}{1-x/(\sqrt{2}-1)} + \frac{\sqrt{2}}{\sqrt{2}+1}\,\frac{1}{1+x/(\sqrt{2}+1)} \\ &= \frac{\sqrt{2}\,(\sqrt{2}+1)}{1-(\sqrt{2}+1)x} - \frac{\sqrt{2}\,(1-\sqrt{2})}{1-(1-\sqrt{2})x}. \end{aligned}$$

By the formula for geometric series, $s_k = \sqrt{2}\,(1+\sqrt{2})^{k+1} - \sqrt{2}\,(1-\sqrt{2})^{k+1}$.

Now we do the problem using the bisection of series idea of the previous paragraph and the result of Exercise 10.1.7. If we call the answer there $a_k(r)$, then we have

$$s_k \;=\; \tfrac{1}{2}\Big(a_k(1/\sqrt{2}) + a_k(-1/\sqrt{2})\Big) \;=\; \tfrac{1}{2}\Big(\sqrt{2}\,(1+\sqrt{2})^{k+1} - \sqrt{2}\,(1-\sqrt{2})^{k+1}\Big).$$

10.1.13. This is multisection with $k=3$ and $j=0,2,1$, respectively. The basic facts that are needed are $e^{i\theta} = \cos\theta + i\sin\theta$ and the sine and cosine of various angles in the 30°-60°-90° right triangle.

10.1.14 (a) By definition

$$N_r \;=\; \sum\Big|S_{i_1}\cap\ldots\cap S_{i_r}\Big|.$$

Thus any object is counted in N_r as many times as we can choose a set of r S_i's each of which contains the object. If an object lies in exactly j of S_i's, the number of times we can make this choice is $\binom{j}{r}$. Partition the objects according to j to obtain the result $(k=j-r)$.

(b) Multiplying by x^r, summing on r:

$$N(x) \;=\; \sum_{j,r} E_j\binom{j}{r}x^r \;=\; \sum_j E_j(1+x)^r.$$

(c) Replace x by $x-1$ in (b) and equate coefficients of x^k in $E(x) = N(x-1)$ to obtain the result.

Section 10.2

10.2.1 (a) Let $a_n = 5a_{n-1} - 6a_{n-2} + b_n$ where $b_1 = 1$ and $b_n = 0$ for $n \neq 1$. Then

$$A(x) = \sum_{k=0}^{\infty}(5xa_{k-1}x^{k-1} - 6x^2a_{k-2}x^{k-2}) + x = 5xA(x) - 6x^2A(x) + x.$$

Thus

$$A(x) = \frac{x}{1-5x+6x^2} = \frac{1}{1-3x} - \frac{1}{1-2x}$$

and $a_n = 3^n - 2^n$.

(b) To correct the recursion, add c_{n+1} to the right side, where $c_0 = 1$ and $c_n = 0$ for $n \neq 0$. Multiply both sides by x^{n+1} and sum to obtain $A(x) = xA(x) + 6x^2A(x) + 1$. With some algebra,

$$A(x) = \frac{1}{1-x-6x^2} = \frac{3/5}{1-3x} + \frac{2/5}{1+2x}$$

and so $a_n = (3^{n+1} - (-2)^{n+1})/5$.

(c) To correct the recursion, add b_n where $b_1 = 1$ and $b_n = 0$ otherwise. $A(x) = xA(x) + x^2A(x) + 2x^3A(x) + x$ and so

$$A(x) = \frac{x}{1-x-x^2-2x^3} = \frac{x}{(1-2x)(1+x+x^2)}.$$

By the quadratic formula, we can factor $1+x+x^2$ as $(1-\omega x)(1-\overline{\omega}x)$, where $\omega = (-1+\sqrt{-3})/2$ and $\overline{\omega}$ is the complex conjugate of ω. Using partial fractions,

$$A(x) = \frac{2/7}{1-2x} - \frac{(3-2\sqrt{-3})/21}{1-\omega x} - \frac{(3+2\sqrt{-3})/21}{1-\overline{\omega}x}$$

and so

$$a_n = \frac{2^{n+1}}{7} - \frac{(3-2\sqrt{-3})\omega^n}{21} - \frac{(3+2\sqrt{-3})\overline{\omega}^n}{21}.$$

The last two terms are messy, but they can be simplified considerably by noting that $\omega^3 = 1$ and so they are periodic with period 3. Thus

$$a_n = \frac{2^{n+1}}{7} + \begin{cases} (-2/7) & \text{if } n/3 \text{ has remainder } 0; \\ 3/7 & \text{if } n/3 \text{ has remainder } 1; \\ (-1/7) & \text{if } n/3 \text{ has remainder } 2. \end{cases}$$

(d) The recursion holds for $n = 0$ as well. From the recursion, $A(x) = 2xA(x) + \sum nx^n$. By Exercise 10.1.5, the sum is $x\frac{d}{dx}\sum x^n$, which is $x/(1-x)^2$. Thus

$$A(x) = \frac{x}{(1-x)^2(1-2x)} = \frac{2}{1-2x} - \frac{1}{1-x} - \frac{1}{(1-x)^2}.$$

After some algebra with these, we obtain $a_n = 2^{n+1} - n - 2$.

10.2.2. We can write $S(n) = 2S(n-1) + 1 + a_n$ for all $n \geq 0$ if we define $S(0) = 0$, $a_0 = -1$, and $a_n = 0$ for $n \neq 0$. Let $s(x) = \sum_{n \geq 0} S(n)x^n$. From the recursion,

$$s(x) = 2xs(x) + \sum_{n \geq 0} x^n - 1 = 2xs(x) + x/(1-x).$$

Thus $s(x) = x/(1-x)(1-2x)$. By partial fractions, $s(x) = (1-2x)^{-1} - (1-x)^{-1}$ and so $S(n) = 2^n - 1$.

10.2.3. Start with a string of $n-i$ zeroes. Choose without repetition i of the $n+1-i$ positions (before all the zeroes or after any zero) and insert a one in each position chosen. The result is an n long string with i ones, none of them adjacent. The process is reversible: The position of a one is the number of zeroes preceding it. The formula for F_n follows immediately.

10.2.4 (a) The elements past the last zero must alternate one and two since there can be no adjacent ones or adjacent twos. There are $n-k$ such terms. The first k elements in the sequence can be anything as long as there are no adjacent ones or twos. Since the k^{th} element is a zero, removing it leads to $k-1$ elements that must satisfy the same conditions. (Ending in zero is critical: If the k^{th} element had been one, removing it would lead to $k-1$ elements with the additional condition that the last element could not be one.)

(b) The sequence can be divided into a first part (up to the last zero) and a last part (after the last zero). If $k \neq 0$, the first part is an arbitrary sequence of the same sort containing $k-1$ terms. If $0 < k < n$, the last part is either $1212\cdots$ or $2121\cdots$ and so there are $2s_{k-1}$ such sequences. If $k = n$, the last part is empty and so there are s_{n-1} such sequences. If $k = 0$, the first part is empty and the last part is an n long sequence of ones and twos. There are 2 such sequences. Putting all this together gives the recursion.

(c) Let $b_0 = 1$, $b_n = 2$ for $n > 0$, $a_0 = 1$, $a_n = s_{n-1}$ for $n > 0$ and $c_n = s_n$. With this definition, $c_n = \sum_{k=0}^{n} a_k b_{n-k}$ for all n. Thus $C(x) = A(x)B(x)$.

(e) $1-2x-x^2 = (1-(1+\sqrt{2})x)(1-(1-\sqrt{2})x)$ and

$$\frac{1+x}{1-2x-x^2} = \frac{(1+\sqrt{2})/2}{1-(1+\sqrt{2})x} + \frac{(1-\sqrt{2})/2}{1-(1-\sqrt{2})x}.$$

Thus $2a_n = (1+\sqrt{2})^{n+1} + (1-\sqrt{2})^{n+1}$.
Checking: This gives $s_0 = 1$, $s_1 = 3$ and $s_2 = 7$, which are correct because we defined $s_0 = 1$, there are clearly three sequences of length 1, and the sequences of length two are all of the 3^2 2-long sequences of $\{0,1,2\}$ except 1,1 and 2,2.

(f) This follows from the previous part since $|1-\sqrt{2}| < 1$.

10.2.5 (a) Replacing A, B and C with their definitions and rearranging leads to

$$L_1L_2 + L_1H_2x^m + L_2H_1x^m + H_1H_2x^{2m} = (L_1 + H_1x^m)(L_2 + H_2x^m).$$

(b) The number of multiplications required by any procedure is an upper bound on $M(2m)$. There are three products of polynomials of degree m or less in our "less direct" procedure. If they are done as efficiently as possible, we will have $M(2m) \leq 3M(m)$.

(c) Let $s_k = M(2^k)$. We have $s_0 = 1$ and $s_k \leq 3s_{k-1}$ for $k > 0$. If we set $t_0 = 1$ and $t_k = 3t_{k-1}$ for $k > 0$, then $s_k \leq t_k$. The recursion gives $T(x) = 3xT(x) + 1$ and so $t_k = 3^k$. Thus, with $n = 2^k$, $M(n) \leq 3^k = (2^{\log_2 3})^k = n^{\log_2 3}$. From tables or a calculator, $\log_2 3 = 1.58\cdots$.

(d) To begin with $L_1(x) = 1+2x$, $H_1(x) = -1+3x$, $L_2(x) = 5+2x$ and $H_2(x) = -x$. The product $L_1L_2 = (1+2x)(5+2x)$ is computed using the algorithm. The values are $m = 1$, $A = (2)(2) = 4$, $B = (1)(5) = 5$ and $C = (1+2)(5+2) = 21$. Thus $L_1L_2 = 5+12x+4x^2$. In a similar way, the products $(-1+3x)(-x) = x-3x^2$ and $(5x)(5+x) = 25x+5x^2$ are computed these are combined to give the final result:

$$(5+12x+4x^2) + (x-3x^2)x^4 + ((25x+5x^2) - (5+12x+4x^2) - (x-3x^2))x^2,$$

which is $5+12x-x^2+12x^3+4x^4+x^5-3x^6$.

(e) We'll just look at the case in which $n = 2m = 2^k$. Let a_k be the number of additions and subtractions needed. We have $a_0 = 0$ and, for $k > 0$, a_k equals $3a_{k-1}$ plus the number of additions and subtractions needed to prepare for and use the three multiplications. Preparation requires two additions of polynomials of degree $m-1$. The results are three polynomials of degree $2m-2$. We must perform two subtractions of such polynomials. Finally, the multiplication by x^m and x^{2m} arranges things so that there is some overlap among the coefficients. In fact, there will be $2m-2$ additions required because of these overlaps (unless some coefficients happen to turn out zero). Since a polynomial of degree d has $d+1$ coefficients, there are a total of

$$2(m-1+1)+2(2m-2+1)+(2m-2) = 4n-4.$$

Thus $a_k = 3a_{k-1}+4\times 2^k-4$ and so $A(x) = 3xA(x)+4\sum_{k>0}(2x)^k-4\sum_{k>0}x^k$. Consequently, $A(x) = 4x/(1-x)(1-2x)(1-3x)$ and $a_k = 2\times 3^{k+1} - 2^{k+3} + 2$. Comparing this with the multiplication result, we see that we need about three times as many additions and/or subtractions as we do multiplications, which is still much smaller than n^2 for large n.

10.2.6 (a) A tree is either a single vertex ($t_1 = 1$) or it has $n+1$ vertices consisting of a root joined to either one n-vertex tree or two trees having a total of n-vertices.

(b) The recursion $t_{n+1} = a_{n+1} + t_n + \sum_{k=0}^{n} t_k t_{n-k}$, where $a_1 = 1$ and $a_n = 0$ otherwise, is valid for all n. Multiply by x^{n+1} and sum to obtain $T(x) = x + xT(x) + xT(x)^2$.

(c) The equation in (b) can be written as $xT^2 - (1-x)T + x = 0$, a quadratic in $T = T(x)$. The result follows from the quadratic formula. As $x \to 0$, $T(x) \to t_0 = 0$. The denominator in the solution approaches zero and the numerator approaches 1 ± 1, thus the minus sign is correct.

Section 10.3

10.3.1 (a) $(1-x)D' - D = -e^{-x} = -(1-x)D$ and so $(1-x)D' - xD = 0$.

(b) The coefficient of x^n on the left of our equation in (a) is

$$\frac{D_{n+1}}{n!} - \frac{D_n}{(n-1)!} - \frac{D_{n-1}}{(n-1)!}.$$

The initial conditions are $D_0 = 1$ and $D_1 = 0$.

10.3.2 (b) Looking at the coefficient of x^n in the differential equation gives

$$(n+1)a_{n+1} - 2na_n - 3(n-1)a_{n-1} = a_n + 3a_{n-1}.$$

Rearrangement leads to the recursion. The initial conditions require that we specify a_0 and a_1, the first two coefficients in the power series for $A(x)$. Thus $a_0 = A(0) = 1$ and $a_1 = A'(0) = 1$.

(c) Following the instructions we have

$$\begin{aligned} A(x) &= (1-(2x+3x^2))^{-1/2} = \sum_{k=0}^{\infty}\binom{-1/2}{k}(-1)^k(2x+3x^2)^k \\ &= \sum_{k=0}^{\infty}\binom{-1/2}{k}(-1)^k\sum_{i=0}^{k}\binom{k}{i}(2x)^i(3x^2)^{k-1} \\ &= \sum_{i,k}\binom{-1/2}{k}(-1)^k\binom{k}{i}2^i3^{k-i}x^{2k-i}. \end{aligned}$$

If we set $2k-i=n$ so that $i=2k-n$, the last sum can be rewritten:

$$A(x) \;=\; \sum_{n,k} \binom{-1/2}{k}(-1)^k \binom{k}{2k-n} 2^{2k-n}3^{n-k}x^n.$$

Thus

$$a_n \;=\; \sum_{k} \binom{-1/2}{k}(-1)^k \binom{k}{2k-n} 2^{2k-n}3^{n-k}.$$

What values of k does the sum range over? In the expansion we obtained for $A(x)$ summing on i and k, these indices could range over all nonnegative integers with $i \le k$ Hence k can range over all nonnegative integers such that $2k-n \le k$. In other words, $k \le n$. The fractional binomial coefficient can be rearranged:

$$(-1)^k\binom{-1/2}{k} \;=\; \frac{(1/2)(1/2+1)\cdots(1/2+(k-1))}{k!} \;=\; \frac{1\cdot 3\cdots(2k-1)}{k!} \;=\; \binom{2k}{k}2^{-k}.$$

We can also write $\binom{k}{2k-n}=\binom{k}{k-(2k-n)}=\binom{k}{n-k}$. Thus we have the nicer looking formula

$$a_n \;=\; \sum_{k=0}^{n}\binom{2k}{k}\binom{k}{n-k}(3/2)^{n-k}.$$

10.3.3 (b) We have $-2\ln(1-x)-2x \;=\; \sum_{k\ge 2} 2x^k/k$ and

$$(1-x)^{-2} \;=\; \sum_{k\ge 0}\binom{-2}{k}(-x)^k \;=\; \sum_{k\ge 0}(k+1)x^k.$$

By the formula for the coefficients in a product of generating functions,

$$\begin{aligned} q_n \;=\; \sum_{k=2}^{n}\frac{2(n-k+1)}{k} &\;=\; 2(n+1)\sum_{k=2}^{n}\frac{1}{k}-\sum_{k=2}^{n}2 \\ &\;=\; 2(n+1)\sum_{k=1}^{n}\frac{1}{k}-2(n+1)-2(n-1) \\ &\;=\; 2(n+1)\sum_{k=1}^{n}\frac{1}{k}-4n. \end{aligned}$$

10.3.4. We multiply by x and differentiate:

$$(xT(x))' \;=\; \frac{-1}{2}+\frac{1+3x}{2\sqrt{1-2x-3x^2}}.$$

Next we multiply by $1-2x-3x^2$:

$$(1-2x-3x^2)(xT(x))' \;=\; \frac{3x^2+2x-1}{2}+(1+3x)\frac{\sqrt{1-2x-2x^2}}{2} \;=\; \frac{(3x^2+2x-1)+(1+3x)(1-x)}{2}-(1+3x)xT(x).$$

Extracting coefficients using $[x^n]\,(xT(x))' \;=\; (n+1)t_n$:

$$(n+1)t_n-2nt_{n-1}-3(n-1)t_{n-2} \;=\; a_n-t_{n-1}-3t_{n-2} \quad\text{where}\quad a_n \;=\; \begin{cases}2, & \text{if } n=1,\\ 0, & \text{otherwise.}\end{cases}$$

Rearranging:

$$t_n \;=\; \frac{(2n-1)t_{n-1}+3(n-2)t_{n-2}+a_n}{n+1}$$

for $n \ge 0$ with the understanding that $t_n=0$ when $n<0$.

Section 10.4

10.4.1 (a) This is nothing more than a special case of the Rule of Product—at each time we can choose anything from $\mathcal{T}$.

(b) Simply sum the previous result on k.

(c) The hint tells how to do it. All that is left is algebra.

(d) The solution is like that in the previous part, except that we start with

$$\prod_{T\in\mathcal{T}}\left(\sum_{i=0}^{\infty}\left(\mathbf{x}^{\mathbf{w}(T)}\right)^i\right) = \prod_{T\in\mathcal{T}}(1-\mathbf{x}^{\mathbf{w}(T)})^{-1}.$$

10.4.2. The rewriting should be fairly obvious. If $A(x)$ is the generating function for the alternating sequences, the Rules of Sum and Product give us $S(x) = A(x) + S(x)xA(x)$. As in Exercise 10.2.4,

$$A(x) \;=\; 1+2x+2x^2+2x^3+\cdots \;=\; 1+\frac{2x}{1-x} \;=\; \frac{1+x}{1-x}.$$

10.4.3 (a) This is simply $2^*\ \{0,\ 12^*\}^*$. Thus the generating function is

$$A(x) = \frac{1}{1-x}\,\frac{1}{1-\left(x+x\dfrac{1}{1-x}\right)} = \frac{1}{1-3x+x^2}.$$

Multiply both sides by $1-3x+x^2$ and equate coefficients of x^n to obtainl the recursion

$$a_n = 3a_{n-1} - a_{n-2} \quad \text{for } n>1$$

with initial conditions $a_0 = 1$ and $a_1 = 3$.

(b) You should be able to see that this is described by $0^*(11^*0^k0^*)^*$. Since

$$\mathsf{G}_{11^*0^k0^*} = x\frac{1}{1-x}x^k\frac{1}{1-x} = \frac{x^{k+1}}{(1-x)^2},$$

the generating function we want is

$$A(x) = \frac{1}{1-x}\,\frac{1}{1-x^{k+1}/(1-x)^2} = \frac{1-x}{1-2x+x^2-x^{k+1}}.$$

Clearing of fractions and equating coefficients, we obtain the recursion

$$a_n = 2a_{n-1} - a_{n-2} + a_{n-k-1} \text{ for } n>1$$

with the understanding that $a_j = 0$ for $j<0$. The initial conditions are $a_0 = a_1 = 1$.

(c) A possible formulation is

$$0^*\ (1(11)^*00^*)^*\ \{\lambda,\ 1(11)^*\}.$$

This says, start with any number of zeroes, then append any number of copies of the patterns of type Z (described soon) and then follow by either nothing or an odd number of ones. A pattern of type Z is an odd number of ones followed by one or more zeroes. The translation to a generating function gives

$$\frac{1}{1-x}\,\frac{1}{\mathsf{G}_Z(x)}\left(1+x\frac{1}{1-x^2}\right) \quad\text{where}\quad \mathsf{G}_Z(x) = x\frac{1}{1-x^2}x\frac{1}{1-x}.$$

After some algebra, the generating function reduces to

$$A(x) = \frac{1+x-x^2}{1-x-2x^2+x^3},$$

which gives $a_n = a_{n-1} + 2a_{n-2} - a_{n-3}$ for $n>2$, with initial conditions $a_0 = 1$, $a_1 = 2$ and $a_2 = 3$.

10.4.4. The generating functions can be worked out just as was done for p_n in the text: Place balls into boxes with the number in the i^{th} box being a multiple of i. If repeated parts are not allowed, the i^{th} box receives either zeor or i balls.

(b) The coefficient of x^n of the right hand side is zero and of the left hand side is $\sum_{k=0}^{n}(-1)^k q_k p_{n-k}$. Rearranging gives

$$\sum_{k=0}^{n/2} q_{2k}p_{n-2k} = \sum_{k=0}^{n/2} q_{2k+1}p_{n-2k-1}.$$

One way to describe this is that if we look at all pairs of partitions such that (i) the first partition has distinct parts and (ii) the sum of all the parts in both partitions is n, then for exactly half of the pairs the first partition is a partition of an odd number. We don't have a direct proof.

(c) The answers are

$$\prod_{i=1}^{k}(1+x^i) \qquad \text{and} \qquad \prod_{i=1}^{k}\frac{1}{1-x^i}.$$

10.4.5. Here's a way to construct a pile of height h. Look at the number of blocks in each column. The numbers increase to h, possibly stay at h for some time, and then fall off. The numbers up to but not including the first h form a partition of a number with largest part at most $h-1$ and the numbers after the first h form a partition of a number with largest part at most h. The structures are these partitions. By the Rule of Product and Exercise 10.4.4

$$\sum_{n\ge 0} s_{n,h}(x) = \left(\prod_{i=1}^{h-1}\frac{1}{1-x_i}\right) x^h \left(\prod_{i=1}^{h}\frac{1}{1-x_i}\right) = \frac{x^h}{(1-x^h)\prod_{i=1}^{h-1}(1-x^i)^2}.$$

Summing this over all $h>0$ and adding 1 gives $\sum s_n x^n$. No simple formula is known for the sum.

10.4.6 (a) Define a map f by $b_1 = a_1$ and $b_i = a_i - a_{i-1}$ for $i>1$. Then $b_i > 0$ for $i \le i \le k$. Note that $b_1 + b_2 + \cdots + b_j = a_j$. Thus the sum of the b_i's does not exceed n. Also, the map f is invertible and any positive k b_j's will give k strictly increasing a_i's. Thus f is a bijection.

(b) Let your jth choice be b_j for $1 \le j \le k$. Then choose the amount by which n will exceed the sum. Keep track of the b_j values and the difference between n and the sum. Adding these together gives n. Since there is exactly one way to make $b_j = i$ for $i>0$ and there is exactly one way to choose the difference between n and the sum to be i for $i \ge 0$, the result follows.

(c) Note that i and a_i have the same parity for all i if and only if all the b_i's are odd. Reasoning as in the previous part we obtain

$$T(x) = \left(\sum_{i\ge 0} x^{2i+1}\right)(1-x)^{-1} = \left(\frac{x}{1-x^2}\right)^k \frac{1}{1-x} = \frac{x^k(1+x)}{(1-x^2)^{k+1}}.$$

(d) Expand the generating function. The coefficient of x^n is $(-1)^j\binom{-k-1}{j}$ where $j=(n-k)/2$ when $n-k$ is even and $j=(n-k-1)/2$ otherwise. Thus $j=\lfloor (n-k)/2\rfloor$. Use $(-1)^j\binom{-k-1}{j} = \binom{k+j}{j} = \binom{k+j}{k}$.

(e) We have a succession for (a_{j-1}, a_j) if and only if $b_j = 1$. Note that $j>1$.

(f) By the binomial theorem applied to $(x+(1-x)y)^{k-1}$, the coefficient of y^j in the previous generating function is the given expression.

(g) The answer is $\binom{k-1}{j}(-1)^t\binom{-k-1+j}{t} = \binom{k-1}{j}\binom{t+k-j}{k-j}$ where $t + 2k - j - 1 = n$. Thus $t = n-2k+j+1$ and so the answer is $\binom{k-1}{j}\binom{n-k+1}{k-j}$. There are 15 subsets the three subsets 1246, 1346 and 1356 each have one succession and the three subsets 1234, 2345 and 3456 each have three successions. The remainder have two successions. Thus, with $k = 4$, we have $s_{6,1} = 3$, $s_{6,2} = 9$, $s_{6,3} = 3$ and all other $s_{n,j}$'s are zero, which agrees with the formula.

10.4.7 (a) We can build the trees by taking a root and joining to it zero, one or two binary RP-trees. This gives us $T(x) = x(1 + T(x) + T(x)^2)$.

(b) There is no simple expansion for the square root; however, various things can be done. One possiblity is to use $\sqrt{1-2x-3x^2} = \sqrt{1-3x}\,\sqrt{1+x}$. You can then expand each square root and multiply the generating functions together. This leads to a summation of about n terms for p_n. The terms alternate in sign. A better approach is to write

$$\sqrt{1-2x-3x^2} = \sum_k \binom{1/2}{k}(-1)^k(2x+3x^2)^k = \sum_{k,j}(-1)^k\binom{1/2}{k}\binom{k}{j}2^{k-j}3^jx^{k+j}.$$

This leads to a summation of about $n/2$ positive terms for p_n. It's also possible to get a recursion by constructing a first order, linear differential equation with polynomial coefficients for $T(x)$ as done in Exercise 10.2.6. Since the recursion contains only two terms, it's the best approach if we want to compute a table of values. It's also the easiest to program on a computer.

10.4.8. Let the number be q_n. We can build the trees by taking a root and using nothing or a binary RP-tree for each of its sons. This gives us $Q(x) = x(1+Q(x))^2$. Solving the quadratic and using $q_0 = 0$: $Q(x) = (1-2x-\sqrt{1-4x})/2x$. Comparing this with the generating function for full binary RP-trees, we see that q_n is the number of full binary RP-trees with $n+1$ leaves when $n > 0$.

There is a simple bijection that proves this equality. If a node has less than two sons, add sons to bring the total to two. This gives a full binary RP-tree. The procedure is clearly reversible. What happens to the number of nodes? All nodes of the original tree become internal nodes in the full tree. Since a full binary RP-tree has one more leaf that internal vertex, we are done.

10.4.9. The key to working this problem is to never allow the root to have exactly one son.

(a) Let the number be r_n. The generating function for those trees whose root has degree k is $R(x)^k$. Since $\sum_{k\ge0} R(x)^k = 1/(1-R(x))$, we have $R(x) = x\frac{1}{1-R(x)} - xR(x)$. Clearing of fractions and solving the quadratic:

$$R(x) = \frac{1+x-\sqrt{1-2x-3x^2}}{2(1+x)}.$$

(The minus sign is the correct choice for the square root because $R(0) = r_0 = 0$.) These numbers are closely related to p_n in Exercise 10.4.7. By comparing the equations for the generating functions,

$$(1+x)R(x) = x(P(x)+1)$$

and so $r_n + r_{n-1} = p_{n-1}$ when $n > 1$.

(b) We modify the previous idea to count by leaves: $R(x) = x+\sum_{k\ge2} R(x)^k = x+R(x)^2/(1-R(x))$. Solving the quadratic:

$$R(x) = \frac{1+x-\sqrt{1-6x+x^2}}{4}.$$

(c) From (a) we have $2(1+x)R - 1 - x = -\sqrt{1-2x-3x^2}$ and so

$$2xR' + 2R - 1 = \frac{1-3x}{\sqrt{1-2x-3x^2}}.$$

Thus $(1-2x-3x^2)(2xR'+R-1) = -2(1+x)R+1+x$. Equating coefficients of x^n gives us

$$(2n+1)r_n - (4n-2)r_{n-1} - (6n-9)r_{n-2} = -2r_n - 2r_{n-1} \quad \text{for } n \geq 3.$$

Rearranging and checking initial conditions we have

$$r_{n+1} = \frac{4nr_n + 3(2n-1)r_{n-1}}{2n+5} \quad \text{for } n \geq 2,$$

with $r_0 = r_2 = 0$ and $r_1 = 1$. You should be able to treat (b) in a similar manner. The result is $r_0 = 0$, $r_1 = r_2 = 1$ and, for $n \geq 2$,

$$r_{n+1} = \frac{3(2n-1)r_n - (n-2)r_{n-1}}{n+1}.$$

10.4.10 (a) By the Rules of Sum and Product and our usual recursive method for building trees,

$$T(x,y) = x + y\sum_{k=1}^{\infty} T(x,y)^k = x + \frac{yT(x,y)}{1-T(x,y)}.$$

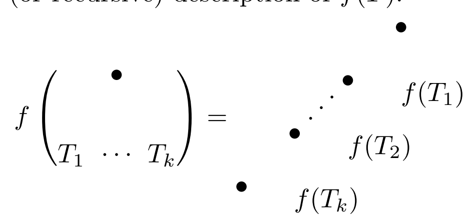
Clearing of fractions and rearranging gives the quadratic $T^2 - (1+x-y)T + x = 0$, which has the solutions

$$T(x,y) = \tfrac{1}{2}\Big(1+x-y \pm \sqrt{1-2x-2y+(x-y)^2}\Big),$$

and $t_{0,0} = 0$ shows that the minus sign is correct.

(b) Since the expression inside square root is unchanged when x and y are interchanged; $U(x,y) = T(x,y) - \frac{1+x-y}{2}$ satisfies $U(x,y) = U(y,x)$. Equating coefficients of $x^n y^k$ gives $u_{n,k} = u_{k,n}$. Since $u_{n,k} = t_{n,k}$ whenever $n+k > 1$, we are done.

(c) Let f be the map. Define $f(\bullet) = \bullet$. Let T be a tree whose sons are the roots of $T_1, \ldots, T_k$. The following picture is a local (or recursive) description of $f(T)$.



10.4.11 (a) A tree of outdegree D, consists of a root and some number $d \in D$ of trees of outdegree D joined to the root. Use the Rules of Sum and Product.

(b) Let $f_0(x) = 1$. Define $f_{n+1}(x) = x\sum_{d\in D} f_n(x)^d$. We leave it to you to prove by induction that $f_n(x)$ agrees with $T_D(x)$ through terms of degree n.

(c) Except for the $1 \in D$ question, this is handled as we did $T_D(x)$. Why must we have $1 \notin D$? You should be able to see that there are an infinite number of trees with exactly one leaf—construct a tree that is just a path of length n from the root to the leaf.

10.4.12. We can place zero balls in $(1,*)$, $(2,*)$, $(3,*)$ in just one way—leave them empty. We can place 5 balls there in five ways, the placements being

$$5,0,0 \quad 3,2,0 \quad 2,0,3 \quad 1,4,0 \quad 0,2,3.$$

Working out similar results for 1, 2, 3 and 4 balls, we get from the hint that the answer is

$$(1+x+2x^2+3x^3+4x^4+5x^5)^n.$$

10.4.13. We can build these trees up the way we built the full binary RP-trees: join two trees at a root. If we distinguish right and left sons, every case will be counted twice, except when the two sons are the same. Thus $B(x) = x + (B(x)^2 - E(x))/2 + E(x)$, where $E(x)$ counts the situation where both sons are the same and nonempty. We get this by choosing a son and then duplicating it. Thus each leaf in the son is replaced by two leaves and so $E(x) = B(x^2)$.

10.4.14 (a) There is a bijection between the sequences and $2n$-long sequences of $(1,0)$ and $(0,1)$ containing an equal number of $(1,0)$ and $(0,1)$ and satisfying one more property to stay above $y=x$. The x coordinate is the number of $(1,0)$'s used and the y coordinate, which must be larger, is the number of $(0,1)$'s. If we replace $(1,0)$ with -1 and $(0,1)$ with $+1$, the difference in coordinates is the sum of the $+1$'s and -1's up to that point.

(b) Call the original sequence type I and the new sequence (end values deleted) type II. We must have $s_1 = 1$ and $s_{2n} = -1$. The generating function for type I sequences is x times the generating function for type II sequences since there is a bijection between type I sequences of length $2n$ and type II sequences of length $2(n-1)$. The partial sums of a type II sequence are nonnegative and the entire sum is 0. Suppose the partial sums are 0 at $k = j_1, j_2, \ldots 2n-1$. Break the sequence into subsequences between s_{j_t} and $s_{j_t}+1$ for all t. The resulting subsequences all sum to 0 and have partial sums strictly positive, thus they are type I. Hence all type II sequences are the juxtaposition of several type I sequences. Translating these ideas into generating functions gives the formula.

(c) We have $S = x/(1-S)$ and so by algebra $S = x + S^2$, the equation for unlabeled full binary RP-trees by leaves and unlabeled RP-tress by vertices. Thus $s_n = b_n$, which we've already computed.

(d) Here is a recursive construction of a bijection $f\colon \mathcal{T}_n \to \mathcal{S}_n$. Suppose that T is an RP-tree and that the sons of the root are $T_1, \ldots, T_k$ in order. (We allow $k=0$, which corresponds to $T = \bullet$.) Let $f(T) = 1, f(T_1), \ldots, f(T_k), -1$. It is easily seen by induction on the number of vertices that the length of $f(T)$ is twice the number of vertices in T. Here is an important observation for later use. If $f(T) = s_1, \ldots, s_m$, then $s_2 + \cdots + s_j \ge 0$, with equality if and only if s_j is the last term in the sequence of one of the $f(T_i)$'s. This observation is easily proved by induction on the length of the sequence.

We now indicate how to prove that f is a bijection. One way to do this is to exhibit a function $g\colon \mathcal{S}_n \to \mathcal{T}_n$ such that $g = f^{-1}$; i.e., $g(f(T)) = T$ for all $T \in \mathcal{T}_n$ and $f(g(S)) = S$ for all $S \in \mathcal{S}_n$. We'll define g recursively. The proof that $g = f^{-1}$ is then done by induction. Let $S = s_1, \ldots, s_{2n}$ be a sequence in $\mathcal{S}_n$. If $n=1$, define $g(S) = \bullet$. Define $j_0 = 1 < j_1 < \cdots < j_t = 2n-1$ by the condition that this sequence contain every $2 < j < 2n$ such that $s_2 + \cdots + s_j = 0$. Let $g(S)$ be a tree whose root has t sons, the kth son being $g(s_{j_{k-1}+1}, \ldots, s_{j_k})$. A key fact in proving $g = f^{-1}$ by induction is the important observation we previously made. Using it, one can show that $g(f(T))$ constructs a tree such that the kth son of the root is $g(f(T_k))$, where T_k is the kth son of the root of T. That allows one to proceed by induction. Similarly, it one can use induction to compute $f(g(S))$.

10.4.15 (a) Either the list consists of repeats of just one item OR it consists of a list of the proper form AND a list of repeats of one item. In the first case we can choose the item in s ways and use it any number of times from 1 to k. In the second case, we can choose the final repeating item in only $s-1$ ways since it must differ from the item preceding it.

(b) After a bit of algebra,

$$A_k(x) \;=\; \frac{s/(s-1)}{1-(s-1)(x+x^2+\cdots+x^k)} - \frac{s}{s-1} \;=\; \frac{s(1-x)/(s-1)}{1-sx+(s-1)x^{k+1}} - \frac{s}{s-1}.$$

(c) Multiplying both sides of the formula just obtained for $A_k(x)$ by $1-sx+(s-1)x^{k+1}$ gives the desired result.

(d) Call a sequence of the desired sort acceptable. Add anything to the end of an n-long acceptable sequence. This gives $sa_{n,k}$ sequences. Each of these is either an acceptable sequence of length $n+1$ or an $(n-k)$-long acceptable sequence followed by $k+1$ copies of something different from the last entry in the $(n-k)$-long sequence.

10.4.16 (a) We have

$$\begin{array}{rl}
0^* & \dfrac{1}{1-x} \\
000^* & \dfrac{x^2}{1-x} \\
1\cup 11 & x+x^2 \\
Z = 000^*(1\cup 11) & \dfrac{x^2}{1-x}(x+x^2) = \dfrac{x^3+x^4}{1-x} \\
Z^* & \dfrac{1}{\frac{x^3+x^4}{1-x}} = \dfrac{1-x}{1-x-x^3-x^4} \\
0^*(1\cup 11)Z^*0^* & \dfrac{x+x^2}{(1-x)^2}\dfrac{1-x}{1-x-x^3-x^4} = \dfrac{x+x^2}{(1-x)(1-x-x^3-x^4)}
\end{array}$$

Thus

$$A(x) = \frac{1}{1-x} + \frac{x+x^2}{(1-x)(1-x-x^3-x^4)} = \frac{1+x^2-x^3-x^4}{(1-x)(1-x-x^3-x^4)} = \frac{1+x+2x^2+x^3}{1-x-x^3-x^4}.$$

(b) The recursion follows from the coefficient of x^n in $(1-x-x^3-x^4)A(x) = 1+x+2x^2+x^3$. The initial conditions are

$$a_0 = 1 \qquad a_1 = 2 \qquad a_2 = 4 \qquad a_3 = 6,$$

which can be obtained from the generating function or by noting that the only sequences of length at most 3 that are forbidden are 101 and 111.

(c) By partial fractions,

$$A(x) = \frac{1}{5}\left(\frac{7+4x}{1-x-x^2} - \frac{2+x}{1+x^2}\right).$$

Since $\frac{1}{x^2} = \sum_{k=0}^{\infty}(-1)^k x^{2k}$, you should be able to complete the derivation of the formula.

(d) We have $F_2 = 1$, $F_3 = 2$ and $F_4 = 3$. For the initial conditions,

$$\begin{array}{lcl}
a_0 = 1 & \text{and} & F_2^2 = 1 \\
a_1 = 2 & \text{and} & F_2F_3 = 2 \\
a_2 = 4 & \text{and} & F_3^2 = 2^2 = 4 \\
a_3 = 6 & \text{and} & F_3F_4 = 2\times 3 = 6.
\end{array}$$

The recursion will be satisfied if

$$F_{n+2}^2 = F_{n+1}F_{n+2} + F_nF_{n+1} + F_n^2 \quad \text{and} \quad F_{n+2}F_{n+3} = F_{n+2}^2 + F_{n+1}^2 + F_nF_{n+1}$$

for $n \geq 0$. We have

$$\begin{aligned}
F_{n+1}F_{n+2} + F_nF_{n+1} + F_n^2 &= F_{n+1}F_{n+2} + F_n(F_{n+1}+F_n) = F_{n+1}F_{n+2} + F_nF_{n+2} \\
&= (F_{n+1}+F_n)F_{n+2} = F_{n+2}^2
\end{aligned}$$

and

$$\begin{aligned}
F_{n+2}^2 + F_{n+1}^2 + F_nF_{n+1} &= F_{n+2}^2 + (F_{n+1}+F_n)F_{n+1} = F_{n+2}^2 + F_{n+2}F_{n+1} \\
&= F_{n+2}(F_{n+2}+F_{n+1}) = F_{n+2}F_{n+3}.
\end{aligned}$$

10.4.17. We have $1-3x+x^2=(1-ax)(1-bx)$ where $a=\frac{3+\sqrt{5}}{2}$ and $b=\frac{3-\sqrt{5}}{2}$. Then

$$\frac{x}{1-3x+x^2}=\frac{1/(a-b)}{1-ax}-\frac{1/(a-b)}{1-bx}$$

and so, since $a-b=\sqrt{5}$,

$$r_n=\frac{a^n-b^n}{\sqrt{5}}.$$

10.4.18 (a) This problem differs from Example 10.17 in one important respect: The vertex 1 in the spanning tree need not have a nice position on the $\mathcal{T}$ tree containing it. In fact, it can be any vertex in that tree except 0. Therefore, we must have one special $\mathcal{T}$-like tree, namely one with the vertex that will become 1 in the spanning tree specially marked. We place this special tree at the start of our list and proceed as indicated.

(b) $\mathsf{G}_{\mathcal{T}'}=\sum k^2x^k=(xd/dx)(xd/dx)\sum x^k=(xd/dx)(x/(1-x)^2)$.

(c) We have the generating function

$$\sum_{k=0}^{\infty}\mathsf{G}_{\mathcal{T}'}(\mathsf{G}_{\mathcal{T}})^k=\frac{\mathsf{G}_{\mathcal{T}'}}{1-\mathsf{G}_{\mathcal{T}}}=\frac{x(1+x)/(1-x)^3}{1-x/(1-x)^2}=\frac{x(1+x)}{(1-x)((1-x)^2-x)}.$$

(d) By partial fractions, the generating function is

$$\frac{2-3x}{1-3x+x^2}-\frac{2}{1-x}.$$

The first fraction is $\frac{2}{x}-3$ times the generating function for r_n.

10.4.19 (a) The accepting states are unchanged except that if the old start state was accepting, both the old and new start states are accepting. If there was an edge from the old start state to state t labeled with input i, then add an edge from the new start state to t labeled with i. (The old edge is *not* removed.) We can express this in terms of the map $f: S\times I\to 2^S$ for the nondeterministic automaton. Let $s_o\in S$ be the old start state and introduce a new start state s_n. Let $T=S\cup\{s_n\}$ and define $f^*: T\times I\to 2^T$ by

$$f*(t,i) \;=\; \begin{cases} f(t,i), & \text{if } t\in S,\\ f(s_o,i), & \text{if } t=s_n.\end{cases}$$

(b) Label the states of $\mathcal{A}$ and $\mathcal{B}$ so that they have no labels in common. Call their start states s_A and s_B. Add a new start state s_n that has edges to all of the states that s_A and s_B did. In other words, $f*(s_n,i)$ is the union of $f_A(s_A,i)$ and $f_B(s_B,i)$, where f_A and f_B are the functions for $\mathcal{A}$ and $\mathcal{B}$. If either s_A or s_B was an accepting state, so is s_n; otherwise the accepting states are unchanged.

(c) Add the start state of $S(\mathcal{A})$ to the accepting states. (This allows the machine to accept the empty string, which is needed since * means "*zero* or more times.") Run edges from the accepting states of $S(\mathcal{A})$ to those states that the start state of $S(\mathcal{A})$ goes to. In other words, if s is the start state,

$$f^*(t,i) \;=\; \begin{cases} f(t,i), & \text{if } t \text{ is not an accepting state,}\\ f(t,i)\cup f(s,i), & \text{if } t \text{ is an accepting state.}\end{cases}$$

(d) From each accepting state of $\mathcal{A}$, run an edge to each state to which the start state of $\mathcal{B}$ has an edge. The accepting states of $\mathcal{B}$ are accepting states. If the start state of $\mathcal{B}$ is an accepting state, then the accepting states of $\mathcal{A}$ are also accepting states, otherwise they are not. The start state is the start state of $\mathcal{A}$.

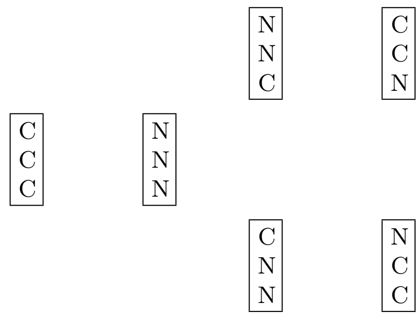


Figure S.11.1 The state transition digraph for covering a 3 by n board with dominoes. Each vertex is labeled with a triple that indicates whether commitment has been made in that row (C) or not made (N). The start and end states are those with no commitments.

Section 11.1

11.1.1. The problem is to eliminate all but the c's from the recursion. One can develop a systematic method for doing this, but we will not since we have generating functions at our disposal. In this particular case, let $p_n = f_n + s_n$ and note that $p_n - p_{n-1} = 2c_{n-1}$ by (11.5). Thus, by the first of (11.4), this result and the last of (11.5),

$$\begin{aligned} c_{n+1} - c_n &= (2c_n + p_n + c_{n-1}) - (2c_{n-1} + p_{n-1} + c_{n-2}) \\ &= 2c_n - c_{n-1} - c_{n-2} + (p_n - p_{n-1}) \\ &= 2c_n + c_{n-1} - c_{n-2}. \end{aligned}$$

11.1.3 (a) Figure S.11.1 gives a state transition digraph.

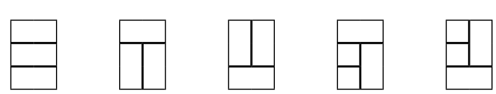
Let $a_{n,s}$ be the the number of ways to take n steps from the state with no commitments and end in a state s. Let $A_s(x) = \sum_n a_{n,s} x^n$. As in the text, the graph lets us write down the linked equations for the generating functions. From the graph it can be seen that A_s depends only on the number k of commitments in s. Therefore we can write $A_s = B_k$. The linked equations are then

$$\begin{aligned} B_0(x) &= x(B_3(x) + 2B_1(x)) + 1 \\ B_1(x) &= x(B_0(x) + B_2(x)) \\ B_2(x) &= xB_1(x) \\ B_3(x) &= xB_0(x), \end{aligned}$$

which can be solved fairly easily for $B_0(x)$.

(b) Equate coefficients of x^n on both sides of $(1 - 4x^2 + x^4)A(x) = 1 - x^2$.

(c) By looking at the dominoes in the last two columns of a board, we see that it can end in five mutually exclusive ways:



This shows that a_n equals $3a_{n-2}$ plus whatever is counted by the last two of the five cases. A board of length $n-2$ ends with either (i) one vertical domino and one horizontal dominoes or (ii) three horizontal dominoes. If the vertical dominoes mentioned in (i) are changed to the left ends of horizontal dominoes, they fit with the last two cases shown above. If the three horizontal mentioned in (ii) are removed, we obtain all boards of length $n-4$. Thus the sum of the last two cases in the picture plus a_{n-4} equals a_{n-2}.

11.1.4. There are 3 choices for the first element of the sequence. For $k > 1$, there are 2 choices for the kth element since it must differ from the $(k-1)$st element. Thus we get $3 \times 2^{n-1}$.

11.1.5. Call the start state α and let $L_{i,j}$ be the number of different single letter inputs that allow the machine to move from state i to state j. Let $a_{n,i}$ be the number of ways to begin in state α, recognize n letters and end in state i and let $A_i = \mathsf{G}^1(a_{n,i})$. The desired generating function is the sum of A_i over all accepting states. A linked set of recursions can be obtained from the automaton that leads to the generating function equations

$$A_i(x) \;=\; \sum_j L_{i,j} x A_j(x) + \begin{cases} 1 & \text{if } i = \alpha; \\ 0 & \text{otherwise.} \end{cases}$$

11.1.6. Use the same vertices as in Example 11.2. Let $b_{n,k}$ be the number of ways to end at state *both* after taking n steps and completing k dominoes. Let $b_n(y) = \sum_k b_{n,k} y^k$. Make similar definitions for c, f and s. On each edge, give the number of dominoes completed in going to the next vertex. The sum of these numbers over a path of length n from *clear* to *clear* gives the number of dominoes on a particular board. Each entry in the array in Example 11.2 will then be a sum of monomials y^d where d is the number of dominoes completed. The systems equations in the example remain valid if we replace the coefficients by the new table:

	clear	*first*	*second*	*both*
clear	$1+y$	1	1	1
first	y	0	y	0
second	y	y	0	0
both	y^2	0	0	0

11.1.7 (a) We will use induction. It is true for $n = 1$ by the definition of $m_{x,y} = m^{(1)}_{x,y}$. (Its also true for $n = 0$ because the zeroth power of a matrix is the identity and so $m^{(0)}_{x,y} = 1$ if $x = y$ and 0 otherwise.) Now suppose that $n > 1$. By the definition of matrix multiplication, $m^{(n)}_{x,y} = \sum_z m^{(n-1)}_{x,z} m_{z,y}$. By the induction hypothesis and the definition of $m_{z,y}$ each term in the sum is the number of ways to get from x to z in $n-1$ steps times the number of ways to get from z to y in one step. By the Rules of Sum and Product, the proof is complete.

(b) If α is the initial state, $\mathbf{i}M^n\mathbf{a}^t = \sum m^{(n)}_{\alpha,y}$, the sum ranging over all accepting states y.

(c) By the previous part, the desired generating function is

$$\sum_{n=0}^{\infty} \mathbf{i}M^n\mathbf{a}^t\, x^n \;=\; \mathbf{i}\sum_{n=0}^{\infty} x^n M^n \mathbf{a}^t \;=\; \mathbf{i}\sum_{n=0}^{\infty}(xM)^n\mathbf{a}^t \;=\; \mathbf{i}(I - xM)^{-1}\mathbf{a}^t.$$

(d) The matrix M is replaced by the table given in the solution to the previous exercise.

11.1.8 (a) Following the hint, there are three vertices called 0, 1 and 2. There are six edges: $(x, 0)$, $(0, x)$ and $(1, 1)$ where $x = 0, 1, 2$. The starting vertex is 0 and all vertices are accepting vertices.

(b) Let $a_{n,k}$ be the number of ways to get from vertex 0 to vertex k in n steps. Then for $n > 0$

$$\begin{aligned} a_{n,0} &= a_{n-1,0} + a_{n-1,1} + a_{n-1,2} \\ a_{n,1} &= a_{n-1,0} + a_{n-1,1} \\ a_{n,1} &= a_{n-1,0}. \end{aligned}$$

With $A_k(x) = \sum_n a_{n,k} x^n$ we have $A_0(x) = x(A_0(x) + A_1(x) + A_2(x)) + 1$, $A_1(x) = x(A_0(x) + A_1(x))$ and $A_2(x) = xA_0(x)$.

(c) Manipulating the last set of equations we have $A_1(x) = xA_0(x)/(1-x)$ and $A_0(x) = xA_0(x)(1+\frac{x}{1-x}+x)+1$. Thus $A_0(x) = (1-x)/(1-2x-x^2+x^3)$ and $A_0(x)+A_1(x)+A_2(x) = A_0(x)(1+\frac{x}{1-x}+x)$.

(d) The solution is $(1,0,0)(I-xM)^{-1}(1,1,1)^t$ where

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

(e) The roots of the equation are roughly 2.2, .55 and $-.8$. By partial fractions, we have that a_n is the closest integer to $\alpha\, r^n$ where

$$r = 2.24697969\ldots \quad \text{and} \quad \alpha = \frac{r(r^2+r-1)}{2r^2+2r-3} = 1.2204108\cdots.$$

(If you're wondering how we got the formula for α, note that it is $(1-rx)(1+x-x^2)/(1-2x-x^2+x^3)$ evaluated at $x = s = 1/r$. By l'Hopital's Rule from calculus, this is $-r(1+s-s^2)/(-2-2s+3s^2)$.)

(f) Let $b_{n,k,s}$ be the number of distributions where we end up in state s, let $B_{n,s}(y) = \sum_k b_{n,k,s}y^k$ and let $B_s(x,y) = \sum_n B_{n,s}(y)x^n$, so y keeps track of the number of balls and x of the number of boxes. Our earlier equations become

$$\begin{aligned} B_{n,0}(y) &= B_{n-1,0}(y) + B_{n-1,1}(y) + B_{n-1,2}(y) \\ B_{n,1}(y) &= y(B_{n-1,0}(y) + B_{n-1,1}(y)) \\ B_{n,1}(y) &= y^2 B_{n-1,0}(y). \end{aligned}$$

Thus $B_0(x,y) = x(B_0(x,y)+B_1(x,y)+B_2(x,y))+1$, $B_1(x,y) = xy(B_0(x,y)+B_1(x,y))$ and $B_2(x,y) = xy^2B_0(x,y)$. Manipulating the last set of equations we have

$$B_0(x,y) = \frac{1-xy}{1-x-xy-x^2y^2+x^3y^3} \quad \text{and} \quad B_0(x,y)+B_1(x,y)+B_2(x,y) = \frac{1+xy^2-x^2y^3}{1-xy}.$$

Section 11.2

11.2.1. Theorem. Suppose each structure in a set $\mathcal{T}$ of structures can be constructed from an ordered partition (K_1, K_2) of the labels, two nonnegative integers ℓ_1 and ℓ_2, and some ordered pair (T_1, T_2) of structures using the labels K_1 in T_1 and K_2 in T_2 such that:

(i) The number of ways to choose a T_i with labels K_i and ℓ_i unlabeled parts depends only on i, $|K_i|$ and ℓ_i.

(ii) Each structure $T \in \mathcal{T}$ arises in exactly one way in this process.

(We allow the possibility of $K_i = \emptyset$ if $\mathcal{T}_i$ contains structures with no labels and likewise for $\ell_i = 0$.) It then follows that

$$T(x,y) \;=\; T_1(x,y)T_2(x,y),$$

where $T_i(x,y) = \sum_{n=0}^{\infty} t_{i,n,m}(x^n/n!)y^m$ and $t_{i,n,m}$ is the number of ways to choose T_i with labels $\underline{n}$ and k unlabeled parts. Define $T(x,y)$ similarly.

The proof is the same as that for the original Rule of Product except that there is a double sum:

$$t_{n,m} = \sum_{K_1 \subseteq \underline{n}} \sum_{\ell_1=0}^{m} t_{1,|K_1|,\ell_1}\, t_{2,n-|K_1|,m-\ell_1} = \sum_{k=0}^{n} \sum_{\ell_1=0}^{m} \binom{n}{k} t_{1,k,\ell_1}\, t_{2,n-k,m-\ell_1}$$

11.2.2 (a) This is nothing more than a special case of the Rule of Product—at each time we can choose anything from $\mathcal{T}$. Repetitions cannot occur because of the labels.

(b) Simply sum the previous result on k.

(c) Sum the result for k-lists on k.

(‘) Since breaking the circular lists at all possible places give all linear lists exactly once, the k long circular lists have generating function $\big(\mathsf{E}_{\mathcal{T}}\big)^k/k$. Sum on k.

11.2.3 (a) By the text,

$$\sum_k z(n,k)y^k = y(y+1)\cdots(y+n-1).$$

Replacing all but the last factor on the right hand side gives us

$$\sum_k z(n,k)y^k = \Big(\sum_k z(n-1,k)y^k\Big)(y+n-1).$$

Equate coefficients of y^k.

(b) For each permutation counted by $z(n,k)$, look at the location of n. There are $z(n-1,k-1)$ ways to construct permutations with n in a cycle by itself. To construct a permutation with n not in a cycle by itself, first construct one of the permutations counted by $z(n-1,k)$ AND then insert n into a cycle. Since there are j ways to insert a number into a j-cycle, the number of ways to insert n is the sum of the cycle lengths, which is $n-1$.

11.2.4. The EGF for the k^{th} box is

$$\sum_{i=0}^{\infty} \frac{x^{ik}}{(ik)!}.$$

11.2.5 (a) For any particular letter appearing an odd number of times, the generating function is

$$\sum_{n \text{ odd}} \frac{x^n}{n!} = \frac{e^x - e^{-1}}{2}$$ with Taylor's theorem and some work.

We must add 1 to this to allow for the letter not being used. The Rule of Product is then used to combine the results for A, B and C.

(b) Multiplying out the previous result:

$$\begin{aligned}\left(1+\frac{e^x-e^{-x}}{2}\right)^3 &= 1+3(e^x-e^{-x})/2+3(e^x-e^{-x})^2/4+(e^x-e^{-x})^3/8\\ &= 1+\Big(3e^x/2-3e^{-x}/2\Big)+\Big(3e^{2x}/4-3/2+3e^{-2x}/4\Big)+\Big(e^{3x}/8-3e^x/8+3e^{-x}/8-e^{-3x}/8\Big)\\ &= -1/2+\Big(e^{3x}/8-e^{-3x}/8\Big)+\Big(3e^{2x}/4+e^{-2x}/4\Big)+\Big(9e^x/8-9e^{-x}/8\Big).\end{aligned}$$

Now compute the coefficients.

11.2.6 (a) Part (a) is essentially the same as before, with 3 replaced by k.

(b) Use the binomial theorem on $(1+z)^k$ where $z=(e^x-e^{-x})/2$.

(c) We can get rid of the generating function, but the result is messy. We'll show that $a_{n,k} = \sum C_{j,k}\, j^n$, where the sum is over all $1 \le j \le k$ such that j and n have the same parity and

$$C_{j,k} \;=\; \sum_{i=0}^{j/2} (-1)^i 2^{1-j-2i} \binom{k}{j+2i} \binom{j+2i}{i}.$$

Write $\sinh(x) = (e^x - e^{-x})/2$. We need the coefficient of $x^n/n!$ in

$$(1+\sinh x)^k \;=\; \sum_{t=0}^{k} \binom{k}{t} (\sinh x)^t.$$

Expanding $(e^x - e^{-x})^t$ by the binomial theorem and using Taylor series for e^z, the coefficient of $x^n/n!$ in $(\sinh x)^t$ is

$$2^{-t} \sum_{i=0}^{t} (-1)^i \binom{t}{i} \big((t-i)-i\big)^n.$$

Note that $i = I$ and $i = t - I$ give the same term in the summation except that the sign changes if $t+n$ is odd. Thus the sum is 0 if $t+n$ is odd and twice the value of the sum over $0 \le i < t/2$ otherwise. Using this fact and changing the index of the first summation from t to $j = t - 2i$, we obtain the desired result.

11.2.7. We saw in this section that $B(x) \;=\; \exp(e^x - 1)$. Differentiating:

$$B'(x) \;=\; \exp(e^x-1)\,(e^x-1)' \;=\; B(x)e^x.$$

Equating coefficients of x^n:

$$\frac{B_{n+1}}{n!} \;=\; \sum_{k=0}^{n} \frac{B_k}{k!}\,\frac{1}{(n-k)!},$$

which gives the result.

11.2.8. By the exponential formula, $A(x) \;=\; \exp(B(x))$ where $B(x)$ is the sum of $x^n/n!$ over odd n. This sum is $(e^x - e^{-x})/2$.

11.2.9 (a) Let $g_{n,k}$ be the number of graphs with n vertices and k components. We have $\sum_{n,k} g_{n,k}(x^n/n!)y^k = \exp(yC(x))$, by the Exponential Formula. Differentiating with respect to y and setting $y=1$ gives us

$$\sum_n \Big(\textstyle\sum_k k g_{n,k}\Big) x^n/n! = \left.\dfrac{\partial \exp(yC(x))}{\partial y}\right|_{y=1} = H(x).$$

(b) $\sum_k g_{n,k}$ is the number of ways to choose an n-vertex graph and mark a component of it. We can construct a graph with a marked component by selecting a component (giving $C(x)$) AND selecting a graph (giving $G(x)$).

(d) Since $C(x) = e^x - 1$, we have $H(x) = (e^x-1)\exp(e^x-1)$ and so

$$h_n \;=\; \sum_{k=1}^{n} \binom{n}{k} B_{n-k} \;=\; \sum_{k=0}^{n} \binom{n}{k} B_{n-k} - B_n,$$

which is $B(n+1) - B_n$ by the previous exercise.

(e) Since $C(x) = x + x^2/2$, we have $H(x) = (x + x^2/2)I(X)$, where $I(x)$ is the EGF for i_n, the number of involutions of $\underline{n}$. Thus the average number of cycles in an involution of $\underline{n}$ is

$$\frac{\binom{n}{1} i_{n-1} + \binom{n}{2} i_{n-2}}{i_n} = \frac{n}{2}\left(1 + \frac{i_{n-1}}{i_n}\right),$$

where the right side comes from the recursion $i_n = i_{n-1} + (n-1)i_{n-2}$.

11.2.10. $A(x) = \sum_{k=0}^{\infty}(e^x - 1)^k = \dfrac{1/2}{1 - e^x/2}$. Expanding this as a geometric series gives $\sum e^{kx}/2^{k+1}$.

11.2.11. Suppose $n > 1$. Since f is alternating,

- k is even;
- $f(1), \ldots, f(k-1)$ is an alternating permutation of $\{f(1), \ldots, f(k-1)\}$;
- $f(k+1), \ldots, f(n)$ is an alternating permutation of $\{f(k+1), \ldots, f(n)\}$.

Thus, an alternating permutation of $\underline{n}$ for $n > 1$ is built from an alternating permutation of odd length AND an alternating permutation, such that the sum of the lengths is $n - 1$. We have shown that

$$\sum_{n>1} \frac{a_n x^{n-1}}{(n-1)!} \;=\; B(x)A(x)$$

and so $A'(x) = B(x)A(x) + 1$. Similarly, $B'(x) = B(x)B(x) + 1$.

Separate variables in $B' = B^2 + 1$ and use $B(0) = 0$ to obtain $B(x) = \tan x$. Use the integrating factor $\cos x$ for

$$A'(x) = (\tan x)A(x) + 1$$

and the initial condition $A(0) = 1$ to obtain $A(x) = \tan x + \sec x$.

11.2.12 (b) Since $S(x) = x(S(x) + 1)^k$, we take $f(u) = (u + 1)^k$ and $g(u) = u$ in Theorem 11.5 and look at the coeffcient of u^{n-1} in $(u + 1)^{kn}/n$, obtaining $t_n = \frac{1}{n}\binom{kn}{n-1}$.

11.2.13 (a) The square of a k-cycle is

- another cycle of length k if k is odd;
- two cycles of length $k/2$ if k is even.

Using this, we see that the condition is necessary. With further study, you should be able to see how to take a square root of such a permutation.

(b) This is simply putting together cycles of various lengths using (a) and recalling that there are $(k-1)!$ k-cycles.

(c) By bisection $\displaystyle\sum_{\substack{k=1 \\ k \text{ odd}}}^{\infty} \frac{x^k}{k} = \tfrac{1}{2}\Big(\{-\ln(1-x)\} - \{-\ln(1-(-x))\}\Big)$.

(d) We don't know of an easier method.

11.2.14 (a) This generating function is

$$\Bigg(\prod_{\substack{k=1 \\ k \text{ odd}}}^{\infty} e^{x^k/k}\Bigg)\Bigg(\prod_{\substack{k=1 \\ k \text{ even}}}^{\infty} e^{yx^k/k}\Bigg) = \exp\Bigg(\sum_{\substack{k=1 \\ k \text{ odd}}}^{\infty} e^{x^k/k}\Bigg)\Bigg\{\exp\Bigg(\sum_{\substack{k=1 \\ k \text{ even}}}^{\infty} e^{x^k/k}\Bigg)\Bigg\}^y$$
$$= \sqrt{\frac{1+x}{1-x}}\Big(\sqrt{(1-x)(1+x)}\Big)^{-y},$$

where the factors in the last equality were obtained by bisection of series $-\ln(1-x) = \sum x^k/k$.

(b) Differentiate the result in (a) with respect to y and set $y = 1$ to get the EGF for the number of even length cycles in a permutation summed over all permutations of a given set:

$$\frac{-\ln(1-x^2)}{2(1-x)}.$$

(c) Let the answer to (b) be a_n and let $b_n = \sum_{k=1}^{n} 1/k$, which is the average number of cycles in a permutation of $\underline{n}$. The requested difference is $(b_n - a_n) - a_n = b_n - 2a_n$. The sum is an approximation to $\int_1^2 x^{-1}dx$ with step size $2/n$.

11.2.15 (a) We'll give two methods. First, we use the Exponential Formula approach in Example 11.13. When we add a root, the number of leaves does not change—except when we started with nothing and ended up with a single vertex tree. Correcting for this exception gives us the formula.

Without the use of the Exponential Formula, we could partition the trees according to the degree k of the root, treating $k = 0$ specially because in this case the root is a leaf:

$$L(x, y) = xy + \sum_{k=1}^{\infty} L(x, y)^k/k!.$$

(c) Differentiate the equation in (a) with respect to y and set $y = 1$ to obtain

$$U(x) = xe^{T(x)}U(x) + x = T(x)U(x) + x,$$

where we have used the fact that $L(x, 1) = T(x) = xe^{T(x)}$. Solving for U: $U = \frac{x}{1-T}$. Differentiating $T(x) = xe^{T(x)}$ and solving for $T'(x)$ gives us $T' = \frac{T}{x(1-T)}$. Thus $x^2T' + x = \frac{x}{1-T}$, which gives the equation for $U(x)$.

We know that $t_n = n^{n-1}$. It follows from the equation for $U(x)$ that

$$\frac{u_n}{n!} = \frac{(n-1)^{n-2}}{(n-2)!}.$$

Thus $u_n/t_n = n/(1 + x)^{1/x}$, where $x = \frac{1}{n-1}$. As $n \to \infty$, $x \to 0$ and, by l'Hôpital's Rule

$$(1 + x)^{1/x} = \exp\left(\frac{\ln(1 + x)}{x}\right) \to \exp(1) = e.$$

11.2.16 (a) Let $\mathcal{T}_n$ be the set of all n-vertex RP-trees. The formula follows from

$$|\mathcal{T}_n| = n^{n-1} \quad \text{and} \quad \sum_{T \in \mathcal{T}_n} h(T) = \sum_k t_{n,k}.$$

(b) By the Exponential Formula (Theorem 11.4 (p. 313)), $e^{H(x,y)}$ counts forests by vertices and sums of heights. When we add a new root, we must increase the heights of each of the vertices by 1. Since x keeps track of all vertices, we can do this simply by replacing x with xy and then add the new root.

(c) Differentiating $T = xe^T$ and doing some algebra, we obtain $xT'(x) = \frac{T(x)}{1-T(x)}$, which will be useful later. We have

$$\begin{aligned} D(x) &= xe^{H(x,1)}\left(x\frac{\partial H(x,1)}{\partial x} + D(x)\right) && \text{by (b)} \\ &= xe^{T(x)}\big(xT'(x) + D(x)\big) \\ &= T(x)\big(xT'(x) + D(x)\big) && \text{by } T = xe^T. \end{aligned}$$

Thus

$$D(x) = \frac{xT'(x)T(x)}{1 - T(x)} = \left(\frac{T(x)}{1 - T(x)}\right)^2.$$

(d) We take $f(u) = e^u$ and we have $g'(u) = 2u/(1-u)^3$. Thus we need the coefficient of u^{n-1} in $2ue^{nu}/(1-u)^3$, which is the same as the coefficient of u^{n-2} in

$$\frac{2}{(1-u)^3} \times e^{nu},$$

which is a product of two easily expanded functions. This gives us

$$2\sum_{k=0}^{n-2} \binom{k+2}{2} \frac{n^{n-2-k}}{(n-2-k)!}$$

for n times the coefficient of x^n in $D(x)$.

11.2.17 (a) There are several steps

- Since g is a function, each vertex of $\varphi(g)$ has outdegree 1. Thus the image of $\underline{n}^{\underline{n}}$ lies in $\mathcal{F}_n$.
- φ is an injection: if $\varphi(g) = \varphi(h)$, then $(x, g(x)) = (x, h(x))$ for all $x \in \underline{n}$ and so $g = h$.
- Finally φ is onto $\mathcal{F}_n$: If $(V, E) \in \mathcal{F}_n$, for each $x \in \underline{n}$ there is an edge $(x, y) \in E$. Define $g(x) = y$.

(b) Let's think in terms of a function g corresponding to the digraph. Let $k \in \underline{n}$. If the equation $g^t(k)$ has a solution, then k is on a cycle and will be the root of a tree. The other vertices of the tree are those $j \in \underline{n}$ for which $g^s(j) = k$ for some s.

(c) This is simply an application of Exercise 11.2.2.

(d) In the notation of Theorem 11.5, $T(x)$ is $T(x)$, $f(t) = e^T$ and $g(u) = -\ln(1-u)$. Thus $n(f_n/n!)$ is the coefficient of u^n in $e^{nu}(1-u)^{-1}$. Using the convolution formula for the coefficient of a product of power series, we obtain the result.

11.2.18. Since $g(u) = (1-4u)(1-3u)^{-2}$, the answer is the coefficient of u^{n-1} in

$$\frac{2(1-6u)}{n(1-3u)^{n+3}},$$

Which equals

$$\frac{2}{n}\left\{(-3)^{n-1}\binom{-n-3}{n-1} - 6(-3)^{n-2}\binom{-n-3}{n-2}\right\}.$$

All that remains is some algebraic manipulation.

Section 11.3

11.3.2. We give the terms in the order the five types of rotations of the cube were listed earlier: no rotation, ±90° F, 180° F, 180° E and ±120° V.

(a) $\frac{1}{24}(x_1^{12} + 6x_4^3 + 3x_2^6 + 6x_1^2x_2^5 + 8x_3^4)$.

(b) $\frac{1}{24}(x_1^8 + 6x_4^2 + 3x_2^4 + 6x_2^4 + 8x_2^2x_3^2)$.

(c) $\frac{1}{24}(x_1^3 + 6x_1x_2 + 3x_1^3 + 6x_1x_2 + 8x_3) = \frac{1}{6}(x_1^3 + 3x_1x_2 + 2x_3)$. This result can also be obtained by noting that the symmetries of the cube induces the group S_3 of all symmetries of $\{x, y, z\}$.

(d) The six labels ($\pm x$, $\pm y$ and $\pm z$) associated with the axes can be associated with the faces of the cube—associate each half-axis with the face it passes through. Thus the answer is the same as the one obtained in the text for faces of the cube.

11.3.3 (a) A regular octahedron can be surrounded by a cube so that each vertex of the octahedron is the center of a face of the cube. The center of the octahedron is the center of the cube. A line segment from the center of the cube to a vertex of the cube passes through the center of the corresponding face of the octahedron. A line segment from the center of the cube to the center of an edge of the cube passes through the center of the corresponding edge of the octahedron.

(b) By the above correspondence, the answer will be the same as the symmetries of the cube acting on the faces of the cube. See (11.31).

(c) By the above correspondence it is the same as the answer for the edges of the cube. See the previous exercise.

11.3.4. Once we have our formula, we can set all v_i's and f_j's equal to 1 to get the result for edges alone. There are two types of axes of rotation of the regular tetrahedron:

- through a vertex and the center of the opposite face with rotations of $\pm 120°$ giving 8 possibilities each yielding a term $v_1 v_3 f_1 f_3 e_3^2$ and
- through the centers of pairs of opposite edges with a rotation of $180°$ giving 3 possibilities each yielding a term $v_2^2 f_2^2 e_1^2 e_2^2$.

In addition, there is the action of doing nothing (identity rotation). Thus we have

$$\frac{v_1^4 e_1^6 f_1^4 + 8 v_1 v_3 f_1 f_3 e_3^2 + 3 v_2^2 f_2^2 e_1^2 e_2^2}{12}.$$

Section 11.4

11.4.1 (a) We have $r^2 = 2r + 1$ and so $r = 1 + \sqrt{2}$ and $m = 1$. By the principle, we expect there is some constant A such that $a_n \sim A(1+\sqrt{2})^n$.

(b) Since $A(x) = \frac{1+x}{1-2x-x^2}$, we have $p(x) = 1 + x$, $q(x) = 1 - 2x - x^2$, $r = \sqrt{2} - 1 = 1/(1+\sqrt{2}\,)$ and $q'(r) = -2\sqrt{2}$. Thus $k = 1$ and we have

$$a_n \sim \frac{(-1)^1 \sqrt{2}\, n^{1-1}}{-2\sqrt{2}\, r^{n+1}} \;=\; \tfrac{1}{2}\big(1+\sqrt{2}\,\big)^{n+1}.$$

(c) We have $1 - 2x - x^2 \;=\; (1-ax)(1-bx)$ where $a = 1 + \sqrt{2}$ and $b = 1 - \sqrt{2}$. Expanding by partial fractions:

$$\begin{aligned}
\frac{1+x}{1-2x-x^2} &= \frac{1}{1-2x-x^2} + \frac{x}{1-2x-x^2} \\
&= \frac{\frac{a}{a-b}}{1-ax} - \frac{\frac{b}{a-b}}{1-bx} \\
&\quad + \frac{\frac{1}{a-b}}{1-ax} - \frac{\frac{1}{a-b}}{1-bx} \\
&= \frac{(2+\sqrt{2}\,)/(2\sqrt{2}\,)}{1-ax} - \frac{(2-\sqrt{2}\,)/(2\sqrt{2}\,)}{1-bx}.
\end{aligned}$$

Thus $a_n \;=\; \frac{1}{2}(1+\sqrt{2}\,)^{n+1} + \frac{1}{2}(1-\sqrt{2}\,)^{n+1}$.

11.4.2. To apply Principle 11.2 (p. 333), we look at $u_n = U_n/n!$ and $v_n = V_n/n!$. The first recursion becomes

$$u_n \;=\; u_{n-1} + \frac{2u_{n-2}}{n(n-1)} + \frac{(n-4)u_{n-3}}{n(n-1)(n-2)} - \frac{u_{n-4}}{n(n-1)(n-2)(n-3)},$$

Which is approximately $u_n = u_{n-1}$ for large n. Since the root is $r = 1$, we expect u_n to behave roughly like 1. The same can be concluded for v_n. Thus we expect U_n and V_n to behave roughly like $n!$.

11.4.3. From the discussion in the example, you can see that merging two lists of lengths i and $j > i$ takes at least i comparison. Thus the example shows that the number of comparisons for merge sorting satisfies $T_n \;=\; f(n) + T(m) + T(n-m)$ where $m = \lfloor n/2 \rfloor$ and $m \le f(n) < n$. Apply Principle 11.3.

11.4.4 (a) Every cycle length d must divide k. Look at the cycle containing n, choosing the other elements in the cycle, arranging them and then choose the remainder of the permutation.

(b) Let $b_n = a_n/(n!)^\alpha$. We must determine α so that the coefficients of the recursion for b_n are asymptotically constant and not all asymptotically 0. Since

$$\frac{\binom{n-1}{d-1}(d-1)!\,a_{n-d}}{(n!)^\alpha} \sim \frac{(n^{d-1}/d)(b_{n-d}(n!/n^d)^\alpha}{(n!)^\alpha} = \frac{n^{d-1-\alpha d}b_{n-d}}{d},$$

we must choose α so that $d-1-\alpha d \le 0$ for all $d|k$, with equality for at least one d. Solving the inequality for α, we find that $\alpha \ge \frac{d-1}{d}$. Since $d \le k$, we must set $\alpha = \frac{k-1}{k}$. The recursion becomes $b_n \sim b_{n-k}/k$. If this were equality, we would have $b_n = C/k^{n/k}$. Thus a_n should grow like $(n!)^{(k-1)/k}/k^{n/k}$, which is roughly like $n!(e/nk)^{1/k}$.

11.4.5. We'll use Principle 11.4 (p. 337) so $t_{n,k}$ will denote the k^{th} term of the sum we're given.

(a) Since

$$\frac{t_{n,k+1}}{t_{n,k}} = \frac{n-k}{n}$$

is less than 1 and is close to 1 when k/n is small, we'll use Principle 11.5 (p. 337). Since

$$\frac{1-r_k}{k} = \frac{1-(n-k)/n}{k} = \frac{1}{n},$$

(11.38) gives the estimate

$$\sqrt{\frac{\pi n}{2}}\,\frac{n!\,n^{n+1}}{n!}.$$

(b) This is a bit more complicated than (a) since

$$\frac{t_{n,k+1}}{t_{n,k}} = \frac{k+1}{k}\,\frac{n-k}{n}$$

is greater than 1 for small k and less than 1 for k near n. Thus $t_{n,k}$ achieves its maximum somewhere between 1 and n, namely, when the above ratio equals 1. This leads to a quadratic equation for k which has the solution

$$k = \frac{-1+\sqrt{1+4n}}{2}.$$

Since this differs from $\sqrt{n}$ by at most a constant, we'll split the sum into two pieces at $k=\sqrt{n}$ and use Principle 11.5 (p. 337) for each half. Since each half has the same estimate, we simply double one result. Ignoring the fact that $\sqrt{n}$ is not an integer, we set $k=\sqrt{n}+j$ and use $j \ge 0$ as the new index of summation. Call the new terms $t'_{n,j}$. We have

$$\begin{aligned}
r_j &= \frac{t'_{n,j+1}}{t'_{n,j}} = \frac{t_{n,k+1}}{t_{n,k}} = \frac{k+1}{k}\,\frac{n-k}{n}\\
&= \frac{\sqrt{n}+j+1}{\sqrt{n}+j}\,\frac{n-\sqrt{n}-j}{n}\\
&= \left(1+\frac{1}{\sqrt{n}+j}\right)\left(1-\frac{\sqrt{n}+j}{n}\right)\\
&= 1+\frac{n-\left(\sqrt{n}+j\right)^2-\left(\sqrt{n}+j\right)}{n\left(\sqrt{n}+j\right)}\\
&= 1-\frac{2j\sqrt{n}+j^2+\sqrt{n}+j}{n\left(\sqrt{n}+j\right)}\\
&\approx 1-\frac{2j\sqrt{n}}{n\sqrt{n}} = 1-2j/n.
\end{aligned}$$

Thus $(1-r_j)/j \approx 2/n$ and so we obtain the following approximation (the factor of 2 is due to the presence of two sums)

$$2\sqrt{\pi n/4}\, t'_{n,0} \;=\; \frac{\sqrt{\pi}\, n!}{n^{\sqrt{n}}(n-\sqrt{n}\,)!},$$

where $(n-\sqrt{n}\,)!$ should be approximated using Stirling's formula (Theorem 1.5 (p. 9)) since we have no formula for $x!$ when x is not a positive integer.

11.4.6 (a) A partition consisting of a single block has EGF $e^x - 1$ since there is just one such partition for each $n > 0$. Thus, the generating function of a list of k blocks is $(e^x-1)^k$. Summing on k gives the result.

(b) In the notation of Principle 11.6 (p. 341), the denominator vanishes at $r = \ln 2$ so we try $f(x) = (1 - x/\ln 2)^{-1}$. Using some algebra and l'Hôpital's Rule,

$$\begin{aligned}\lim_{x\to\ln 2}\frac{A(x)}{f(x)} &= \lim_{x\to\ln 2}\frac{1-x/\ln 2}{2-e^x}\\ &= \lim_{x\to\ln 2}\frac{-1/\ln 2}{-e^x} \;=\; \frac{1}{2\ln 2}.\end{aligned}$$

Remembering that we have an EGF, (11.43) gives

$$\frac{a_n}{n!} \sim \frac{1}{2\ln 2}\binom{n-(-1)-1}{n}(\ln n)^0(\ln 2)^{-n} \;=\; \frac{1}{2(\ln 2)^{n+1}}.$$

(c) Since $A(x) = \frac{1}{2}(1-e^x/2)^{-1} = \frac{1}{2}\sum(1/2)^k e^{kx}$, the summation for a_n follows easily. Using $1+x \approx e^x$ for small x, we have

$$\frac{t_{n,k+1}}{t_{n,k}} \;=\; \frac{(k+1)^n}{2k^n} \;=\; \tfrac{1}{2}(1+1/k)^n \;\sim\; e^{n/k}/2.$$

Thus the maximum term occurs at about $k = n/\ln 2$. Split into two sums at this point and use Principle 11.5 (p. 337). Both sums will be asymptotically equal. Let $k = n/\ln 2 + j$.

$$\begin{aligned}\frac{t'_{n,j+1}}{t'_{n,j}} &= \tfrac{1}{2}\left(1+\frac{1}{j+n/\ln 2}\right)^n \;\sim\; \tfrac{1}{2}\exp\left(\frac{n}{j+n/\ln 2}\right)\\ &= \exp\left(\frac{n}{j+n/\ln 2} - \ln 2\right) \;=\; \exp\left(\frac{-j\ln 2}{n/\ln 2 + j}\right)\\ &\sim \exp\Big(-j(\ln 2)^2/n\Big) \;\sim\; 1 - j(\ln 2)^2/n.\end{aligned}$$

Thus $(1-r_j)/j \sim (\ln 2)^2/n$ and so we obtain

$$a_n \sim 2\sqrt{\pi n/2(\ln 2)^2}\, t_{m,n/\ln 2} \;=\; \frac{\sqrt{2\pi n}\,(n/\ln 2)^n}{\ln 2\; 2^{n/\ln 2+1}} \;=\; \frac{1}{2(\ln 2)^{n+1}}\,\frac{\sqrt{2\pi n}\, n^n}{e^n}.$$

This differs from the first estimate by an application of Stirling's formula for $n!$.

11.4.7. Use Principle 11.6 (p. 341) with $r = 1$, $b = 0$ and $c = -1$ to obtain

$$a_n \sim n!\,\exp\Big(-\sum_{k\in S} 1/k\Big).$$

11.4.8 (a) The number of undirected k-cycles on $\underline{k}$ is 1 if $k=1$ or $k=2$ and is $(k-1)!/2$ if $k>2$ because there are $(k-1)!$ *directed* k-cycles and each undirected cycle gives rise to two directed ones if $k>2$. However, the 1-cycles and 2-cycles are not allowed in a simple graph since a 1-cycle gives a loop and a 2-cycle gives a double edge. Hence the number is $(-1)!/2$ for $k>2$ and 0 for $k\le 2$. Use the Exponential Formula (Theorem 11.4 (p. 313)).

(b) We use Principle 11.6 (p. 341) with $r=1$, $b=0$ and $c=-1/2$. Since $L=\exp(-\frac{1}{2}-\frac{1}{4})=e^{-3/4}$,

$$a_n \sim n!\,e^{-3/4}n^{-1/2}/\Gamma(1/2) = \frac{n!}{e^{3/4}\sqrt{\pi n}}.$$

11.4.9 (a) The functions have singularities at $x=-1$ as well as at $x=1$, so we can't use Principle 11.6 (p. 341). Another reason we can't use it for $A_e(x)$ is that $a_{e,n}=0$ whenever n is odd.

(b) For both cases, $r=1$, $b=0$ and $c=-1/2$. We obtain $L=\sqrt{2}$ for $A_o(x)$ and $L=1/\sqrt{2}$ for $A_e(x)$.

(c) By power series, $a_{e,2n}=(-1)^n\binom{-1/2}{n}(2n)!$, which can be rearranged to give the answer. By Stirling's formula, $a_{e,2n}\sim(2n)!/\sqrt{\pi n}\sim 2(2n/e)^{2n}$.

(d) Since $A_o(x)=(1+x)A_e(x)$, we have $a_{o,2n}=a_{e,2n}$ and $a_{o,2n+1}=(2n+1)a_{e,2n}$. By the previous part, $a_{o,2n}=\binom{2n}{n}(2n)!\,4^{-n}$ and $a_{o,2n+1}=\binom{2n}{n}(2n+1)!\,4^{-n}$.

11.4.10 (a) This follows easily from Exercise 10.4.1(b).

(b) Since S is finite, $\sum_{k\in S}x^k$ is a polynomial and so we have a rational generating function. We can use Example 11.28.

(c) Again, the generating function is rational, but it is not so obvious:

$$\frac{1}{1-\sum_{k\in S}x^k} = \frac{1}{1-\left(\sum_{t=1}^{\infty}x^t-\sum_{k\in S'}x^k\right)} = \frac{1}{1-\frac{x}{1-x}+\sum_{k\in S'}x^k} = \frac{1-x}{1-2x+(1-x)\sum_{k\in S'}x^k}.$$

11.4.11. We use Principle 11.6 (p. 341) with

$$A(x) = (1-2x-3x^2)^{-1/2},\quad b=0 \text{ and } c=-1/2.$$

Since $1-2x-3x^2=(1-3x)(1+x)$, $r=1/3$ and

$$L = \lim_{x\to 1/3}\frac{(1-2x-3x^2)^{-1/2}}{(1-3x)^{-1/2}} = \lim_{x\to 1/3}\frac{1}{\sqrt{1+x}} = \frac{\sqrt{3}}{2}.$$

Thus

$$a_n \sim \frac{\sqrt{3}\,3^n n^{-1/2}}{2\Gamma(1/2)} = \frac{3^{n+1/2}}{2\sqrt{\pi n}}.$$

11.4.12. For $n>0$, all that matters is $-\sqrt{1-2x-3x^2}\,/2x$. This can be dealt with much like the generating function in the previous problem.

11.4.13. We use Principle 11.6 (p. 341) with

$$A(x) = \frac{-\sqrt{(1+x^2)^2-4x}}{2},\quad b=0 \quad\text{and}\quad c=1/2.$$

The square root vanishes at $x = r = 0.295597742\ldots$ and so

$$L \;=\; \lim_{x\to r} \frac{-\sqrt{(1+x^2)^2-4x}}{2\sqrt{1-x/r}} \;=\; -\frac{1}{2}\sqrt{\lim_{x\to r}\frac{(1-x^2)^2-4x}{1-x/r}}.$$

By using l'Hôpital's Rule, we obtain

$$L \;=\; -\frac{1}{2}\sqrt{\frac{-4r(1-r^2)-4}{-1/r}} \;=\; -\sqrt{r+r^2(1-r^2)} \;=\; -0.61265067.$$

11.4.14. We give two methods for obtaining the generating function.

The generating function for one cycle is $\sum(n-1)!\,x^n/n! = -\ln(1-x)$. An ordered k-list is obtained by taking the k^{th} power and it is unordered by dividing by $k!$ since labeled objects will always have different labels.

Use the Exponential Formula (Theorem 11.4 (p. 313)) with a second variable keeping track of the number of cycles to get $\exp(-y\ln(1-x))$. Extract the coefficient of y^k.

For the asymptotics, use Principle 11.6 (p. 341) with $r = 1$, $b = k$ and $c = 0$. Then $L = 1/k!$ and

$$a_n \sim \frac{n!}{k!}\,\frac{k(\ln n)^{k-1}}{n} \;=\; \frac{(n-1)!\,(\ln n)^{k-1}}{(k-1)!}.$$

11.4.15. By techniques we have used before,

$$H(x) \;=\; x\sum_{k\ge 2} H(x)^k + x.$$

Sum the geometric series and use algebra to obtain the desired quadratic equation for $H(x)$.

This quadratic could be treated as an implicit equation for $H(x)$ and we could apply Principle 11.7 (p. 345). Alternatively, we could solve the quadratic for $H(x)$ and use Principle 11.6 (p. 341). For the former, let $F(x,y) = y^2 - y + \frac{x}{1+x}$. For the latter,

$$H(x) \;=\; \frac{1-\sqrt{1-4x/(1+x)}}{2}$$

So we take $A(x) = -\sqrt{1-4x/(1+x)}/2$, $r = 1/3$, $b = 0$ and $c = 1/2$. In any case, the answer is

$$a_n \;\sim\; \frac{\sqrt{3}\,3^n}{4\sqrt{\pi n^3}}.$$

11.4.16. We use Principle 11.7 (p. 345) with

$$F(x,y) \;=\; (1+x)y - xe^y, \quad F_y(x,y) \;=\; 1+x-xe^y \quad \text{and} \quad y \;=\; H(x).$$

From $F(r,s) = 0 = F_y(r,s)$, we see that $s = 1$ and so $1 + r - re = 0$, which gives $r = \frac{1}{e-1}$. Easily

$$F_x(r,s) \;=\; s - e^s = 1-e \quad \text{and} \quad F_{yy}(r,s) \;=\; -re^s \;=\; \frac{-e}{e-1}.$$

Thus $h_n/n! \;\sim\; (e-1)^{n+1/2}/\sqrt{2\pi e n^3}$.

11.4.17. We would like to use Principle 11.6 (p. 341), but there is a problem with all of the principles about when we can use them. At any rate, we want to solve

$$\sum_{k\in D} r^k/k! \;=\; 1$$

for r. This can always be done since the sum vanishes at $r=0$ and goes to $+\infty$ as $r\to+\infty$.

Let $d=\gcd(D)$. You should be able to see that $a_n=0$ when n is not a multiple of d. Hence we'll need to assume that $d=1$. (Actually you can get around this by setting $x^d=z$, a new variable.)

It turns out that when all this is done, (11.43) gives the correct answer, where r is as indicated, $b=0$ and $c=-1$. The answer for general d is

$$a_n \;\sim\; \frac{d\,n!\,r^{-n}}{\sum_{k\in D} r^k/(k-1)!} \quad \text{when } d \text{ divides } n$$

and $a_n=0$, otherwise.

11.4.18. Use Principle 11.7 (p. 345) as in Example 11.33. We have

$$F(x,y) \;=\; 1-y+x(y^2+T(x^2))/2 \quad\text{and}\quad F_y(x,y) \;=\; -1+xy.$$

Thus $s=1/r$ and, using this in $F(r,s)=0$ followed by some algebra,

$$r^2T(r^2)=1-2r.$$

To evaluate $xT(x)$ for particular x, we first solve the quadratic to obtain

$$xT(x) \;=\; \frac{1-\sqrt{1-2x-x^2T(x^2)}}{x} \;=\; \frac{2x+x^2T(x^2)}{1+\sqrt{1-2x-x^2T(x^2)}}.$$

This can be iterated to express $xT(x)$ in terms of x and $x^{2^k}T(x^{2^k})$. Since $x^{2^k}T(x^{2^k})$ rapidly approaches 1 as k grows, we can obtain good estimates for $xT(x)$. In this way, we obtain $r=0.402697\ldots$. We omit the details.

11.4.19. Use Principle 11.6 (p. 341) in this exercise.

(a) Set $f(x)=(1-x/r)^{ck}$ and note that

$$\frac{B(x)}{f(x)} \;=\; \left(\frac{A(x)}{(1-x/r)^c}\right)^k \to L^k \quad\text{as}\quad x\to r.$$

(b) We have

$$\Big(g(x)+p(x)\Big)^k \;=\; p(x)^k+\sum_{i=1}^{k}\binom{k}{i}p(x)^{k-i}g(x)^i.$$

Look at each term on the right side separately. Since $p(x)^k$ is a polynomial, it does not contribute to the asymptotics. It is also possible that $g(x)^i$ is a polynomial for some $i>1$ and so that term does not contribute to the asymptotics either. For those that do contribute, we have

$$\lim_{x\to r}\frac{\binom{k}{i}p(x)^{k-i}g(x)^i}{(1-x/r)^{ci}} \;=\; \binom{k}{i}p(r)^{k-i}L^i,$$

so the contribution should be asymptotic to

$$\left\{ \binom{k}{i} p(r)^{k-i} L^i / \Gamma(-ci) \right\} \; n^{-ci-1} \, r^{-n}.$$

The first factor is independent of n and the last factor is independent of i so the relative importance of the terms is determined by n^{-ci-1}, which is largest when i is as small as possible; i.e., $i = 1$. This gives the result.

(c) Since $A(x)$ is a sum of nonnegative terms, it is an increasing function of x and so $A(x) = 1$ has at most one positive solution. We take $b = 0$, $c = -1$ and $f(x) = (1 - x/s)^{-1}$ in Principle 11.6. Then

$$\lim_{x \to s} \frac{(1 - A(x))^{-1}}{(1 - x/s)^{-1}} \;=\; \lim_{x \to s} \frac{1 - x/s}{1 - A(x)} \;=\; \frac{1}{sA'(s)}$$

by l'Hôpital's Rule.

Suppose that $c < 0$. Note that $A(0) = 0$ and that $A(x)$ is unbounded as $x \to r$ because $A(x)/(1 - x/r)^c$ approaches a nonzero limit. Thus $A(x) = 1$ has a solution in $(0, r)$.

11.4.20. ???

11.4.21. For Exercise 11.2.2(a), use Exercise 11.4.19(a,b).
For Exercise 11.2.2(b), use Exercise 11.4.19(c).
Exercise 11.2.2(d) can be done like Exercise 11.4.19(c) was.
The only difference is that, since we are dealing with a logarithm, Principle 11.6 is used with $b = 1$ and $c = 0$ instead of with $b = 0$ and $c = -1$.

11.4.22 (b) Let $U(x) = T(x)/x$. The equation for U is $U = \sum x^d U^d / d!$. Replacing x^k with z, we see that this leads to a power series for U in powers of $z = x^k$. Thus the coefficients of x^m in $U(x)$ will be 0 when m is not a multiple of k.

(c) We apply Principle 11.7 (p. 345) with

$$F(x,y) \;=\; y - x \sum_{d \in D} y^d / d! \quad \text{and} \quad F_y(x,y) = 1 - x \sum_{\substack{d \in D \\ d \neq 0}} y^{d-1}/(d-1)!.$$

Using $F(r,s) = 0 = F_y(r,s)$ and some algebra, we obtain

$$\sum_{\substack{d \in D \\ d \neq 0}} (d-1) s^d / d! \;=\; 1 \quad \text{and} \quad r \;=\; \left(\sum_{\substack{d \in D \\ d \neq 0}} s^{d-1}/(d-1)! \right)^{-1}.$$

Once the first of these has been solved numerically for s, the rest of the calculations are straightforward.

Appendix A

A.1. $\mathcal{A}(n)$ is the formula $1+3+\cdots+(2n-1)=n^2$ and $n_0=n_1=1$. A(1) is just $1=1^2$. To prove $\mathcal{A}(n+1)$ in the inductive step, use $\mathcal{A}(n)$:

$$(1+3+\cdots+(2n-1))+(2n+1)=n^2+(2n+1)=(n+1)^2.$$

A.2. $\mathcal{A}(n)$ is the equality and $n_0=n_10$. It is simple to check that $\mathcal{A}(0)$ is true. Now for the inductive step with $n>0$.

$$\begin{aligned}\sum_{k=0}^{n}k^2 &= n^2+\sum_{k=0}^{n-1}k^2\\ &= n^2+(n-1)(n-1+1)(2(n-1)+1)/6 && \text{by } \mathcal{A}(n-1)\\ &= n(n+1)(2n+1)/6 && \text{by algebra.}\end{aligned}$$

A.3. Let $\mathcal{A}(n)$ be $\sum_{k=1}^{n-1}(-1)^k k^2 \;=\; (-1)^{n-1}\sum_{k=1}^{n}k$. By (A.1), we can replace the right hand side of $\mathcal{A}(n)$ by $(-1)^{n-1}n(n+1)/2$, which we will do. It is easy to verify $\mathcal{A}(1)$. As usual, the induction step uses $\mathcal{A}(n-1)$ to evaluate $\sum_{k=1}^{n-1}(-1)^{k-1}k^2$ and some algebra to prove $\mathcal{A}(n)$ from this.

What would have happened if we hadn't thought to use (A.1)? The proof would have gotten more complicated. To prove $\mathcal{A}(n)$ we would have needed to prove that

$$(-1)^{(n-1)-1}\sum_{k=1}^{n-1}k+(-1)^{n-1}n^2 \;=\; (-1)^{n-1}\sum_{k=1}^{n}k.$$

At this point, we would have to prove this result separately by induction or prove in using (A.1).

A.4. Let $\mathcal{A}(n)$ be

$$\sum_{k=0}^{n-1}\frac{2^k x^{2^k}}{1+x^t k}=\frac{x}{1-x}-\frac{2^n x^{2^n}}{1-x^{2^n}}.$$

For $\mathcal{A}(1)$, the left side is $x/(1+x)$ and the right side is

$$\frac{x}{1-x}-\frac{2x^2}{1-x^2}=\frac{x(1+x)-2x^2}{(1-x)(1+x)}=\frac{x}{1+x}.$$

The inductive step follows the usual pattern we've established for sums, but there's a bit of algebra:

$$\begin{aligned}\sum_{k=0}^{n-1}\frac{2^k x^{2^k}}{1+x^{2^k}} &= \sum_{k=0}^{n-2}\frac{2^k x^{2^k}}{1+x^{2^k}}+\frac{2^{n-1}x^{2^{n-1}}}{1+x^{2^{n-1}}}\\ &= \frac{x}{1-x}-\frac{2^{n-1}x^{2^{n-1}}}{1-x^{2^{n-1}}}+\frac{2^{n-1}x^{2^{n-1}}}{1+x^{2^{n-1}}}\\ &= \frac{x}{1-x}-2^{n-1}x^{2^{n-1}}\,\frac{1+x^{2^{n-1}}-(1-x^{2^{n-1}})}{(1-x^{2^{n-1}})(1+x^{2^{n-1}})}\\ &= \frac{x}{1-x}-2^{n-1}x^{2^{n-1}}\,\frac{2x^{2^{n-1}}}{1-x^{2^n}}.\end{aligned}$$

A.5. The claim is true for $n=1$. For $n+1$, we have

$$(x^{n+1})' \;=\; (x^n x)' \;=\; (x^n)'x+(x^n)x'=(nx^{n-1})x+x^n,$$

where the last used the induction hypothesis. Since the right side is $(n+1)x^n$, we are done.

A.6. Call the answer I_n. Note that $I_0 = \int_0^\infty e^{-x}dx = 1$. When $n > 0$, we can evaluate the integral using integration by parts with $u = x^n$ and $dv = e^{-x}dx$:

$$\int_0^\infty x^n e^{-x}dx \;=\; x^n(-e^{-x})\Big|_0^\infty + \int_0^\infty nx^{n-1}e^{-x}dx \;=\; 0 + nI_{n-1}.$$

You should be able to see that $I_n = n!$.
Now for the proof. The formula is true for $n = 0$ since we calculated that $I_0 = 1$. For $n > 0$ we have $I_n = nI_{n-1} = n(n-1)! = n!$, where we used the induction hypothesis to evaluate I_{n-1}.

A.7. The inductive step only holds for $n \geq 3$ because the claim that P_{n-1} belongs to both groups requires $n - 1 \geq 2$; however, $\mathcal{A}(2)$ was never proved. (Indeed, if $\mathcal{A}(2)$ is true, then $\mathcal{A}(n)$ is true for all n.)

A.8. $\mathcal{A}(1)$, which was never checked, is false.

A.9. This is obviously true for $n = 1$. Suppose we have a numbering when $n - 1$ lines have been drawn. The n^{th} line divides the plane into two parts, say A and B. Assign all regions in A the same number they had with $n - 1$ lines and reverse the numbering in B.

Section B.1

B.1.1. We'll do the case where the functions need not be nonnegative. Also, the proofs for O properties are omitted because they are like those for Θ without the "A" part of the inequalities. Throughout for f and g (or f_i and g_i) let A and B (or A_i and B_i) be as in the definition of Θ. All inequalities are understood to hold for some A's and B's and all sufficiently large n.

(a) Since $g(n)$ is $\Theta(f(n))$, $|g(n)| \leq B|f(n)|$, which means that $g(n)$ is $\Theta(f(n))$.

(b) This follows from the definition with $A = B = 1$.

(c) We need not require that C and D be positive. Let $A' = A|C/D|$ and $B' = B|C/D|$. Then $A'|Df(n)| \leq |Cg(n)| \leq B'|Df(n)|$.

(d) We have $(1/B)|g(n)| \leq |f(n)| \leq (1/A)|g(n)|$.

(e) This was done in the text.

(f) As noted in the remark, this requires that f_i and g_i ben nonnegative. With $A = \min(A_1, A_2)$, $B = B_1 + B_2$ and $f(n) = \max(f_1(n), f_2(n))$ we have

$$Af(n) \leq \max(A_1f_1(n), A_2f_2(n)) \leq g_1(n) + g_2(n) \leq Bf(n).$$

B.1.2. Let $g_1(n) = -g_2(n) = n$ and $f_1(n) = f_2(n) = n$. Then $g_i(n)$ is $\Theta(f_i(n))$ but $g_1(n)+g_2(n) = 0$ and $\max(f_1(n), f_2(n)) = n$.

B.1.3. This is not true. For example, n is $O(n^2)$, but n^2 is not $O(n)$.

B.1.4 (a) Hint: You can first show this with $g(x) = x^3$ and adjust the constants.

(b) Hint: You can first show this with $g(x) = x^2$ and adjust the constants.

B.1.5 (a) Hint: There is an explicit formula for the sum of the squares of integers.

(b) Hint: There is an explicit formula for the sum of the cubes of integers.

(c) Hint: If you know calculus, upper and lower Riemann sum approximations to the integral of $f(x) = x^{1/2}$ can be used here.

B.1.6 (a) Hint: If you know calculus, upper and lower Riemann sum approximations to the integral of $f(x) = x^{-1}$ can be used here.

(b) Hint: If you know calculus, upper Riemann sum approximaitions to the integral of $f(x) = \log_b(x)$ can be used here.

B.1.7 (a) Here's a chart of values.

	5	10	30	100	300
n^2	25	10^2	9×10^2	10^4	9×10^4
$100n$	5×10^2	10^3	3×10^3	10^4	3×10^4
$100(2^{n/10} - 1)$	41	10^2	7×10^2	10^5	10^8
fastest	A	$A,\ C$	C	$A,\ B$	B
slowest	B	B	B	C	C

(b) When n is very large, B is fastest and C is slowest. This is because, (i) of two polynomials the one with the lower degree is eventually faster and (ii) an exponential function grows faster than any polynomial.

B.1.8. **Need solution**

B.1.9. Let $p(n) = \sum_{i=0}^{k} b_i n^i$ with $b_k > 0$.

(a) Let $s = \sum_{i=0}^{k-1} |b_i|$ and assume that $n \geq 2s/b_k$. We have

$$|p(n) - b_k n^k| \leq \left| \sum_{i=0}^{k-1} b_i n^i \right| \leq \sum_{i=0}^{k-1} |b_i| n^i \leq \sum_{i=0}^{k-1} |b_i| n^{k-1} = s n^{k-1} \leq b_k n^k/2.$$

Thus $|p(n)| \geq b_k n^k - b_k n^k/2 \geq (b_k/2) n^k$ and also $|p(n)| \leq b_k n^k + b_k n^k/2 \leq (3b_k/2) n^k$.

(b) This follows form (a) of the theorem.

(c) By applying l'Hospital's Rule k times, we see that the limit of $p(n)/a^n$ is $\lim_{n\to\infty} (k!/(loga)^k)/a^n$, which is 0.

(d) By the proof of the first part, $p(n) \leq (3b_k/2)n^k$ for all sufficiently large n. Thus we can take $C \geq 3b_k/2$.

(e) For $p(n)$ to be $\Theta(a^{Cn^k})$, we must have positive constants A and B such that $A \leq a^{p(n)}/a^{Cn^k} \leq B$. Taking logarithms gives us $\log_a A \leq p(n) - Cn^k \leq \log_a B$. The center of this expression is a polynomial which is not constant unless $p(n) = Cn^k + D$ for some constant D, the case which is ruled out. Thus $p(n) - Cn^k$ is a nonconstant polynomial and so is unbounded.

B.1.10. We have $a^{f(n)}/b^{f(n)} = (a/b)^{f(n)} \to 0$ since $0 < a/b < 1$ and $f(n) \to +\infty$. Also, $a^{g(n)}/a^{f(n)+g(n)} = a^{-f(n)} \to 0$.

B.1.11 (a) The worst time possibility would be to run through the entire loop because the "`If`" always fails. In this case the running time is $\Theta(n)$. This actually happens for the permutation $a_i = i$ for all i.

(b) Let N_k be the number of permutations which have $a_{i-1} < a_i$ for $2 \leq i \leq k$ and $a_k > a_{k+1}$. (There is an ambiguity about what to do for the permutation $a_i = i$ for all i, but it contributes a negligible amount to the average running time.) The "`If`" statement is executed k times for such permutations. Thus the average number of times the "`If`" is executed is $\sum kN_k/n!$. If the a_i's were chosen independently one at a time from all the integers so that no adjacent ones are equal, the chances that all the k inequalities $a_1 < a_2 < \cdots < a_k > a_{k+1}$ hold would be $(1/2)^k$.

This would give $N_k/n! = (1/2)^k$ and then $\sum_{k=0}^{\infty} kN_k/n!$ would converge by the "ratio test." This says that the average running time is bounded for all n. Unfortunately the a_i's cannot be chosen as described to produce a permutation of $\underline{n}$.

We need to determine N_k. With each arbitrary permutation $a_1, a_2, \ldots$ we can associate a set of permutations $b_1, b_2, \ldots$ counted by N_k. We'll call this the set for $a_1, a_2, \ldots$. For $i > k+1$, $b_i = a_i$, and $b_1, \ldots, b_{k+1}$ is a rearrangement of $a_1, \ldots, a_{k+1}$ to give a permutation counted by N_k. How many such rearrangements are there? b_{k+1} can be any but the largest of the a_i's and the remaining b_i's must be the remaining a_i's arranged in increasing order. Thus there are k possibilities and so the set for $a_1, a_2, \ldots$ has k elements. Hence the set associated with $a_1, a_2, \ldots$ contains k permutations counted by N_k. Since there are $n!$ permutations, we have a total of $n!k$ things counted by N_k; however, each permutation $b_1, b_2, \ldots$ counted by N_k appears in many sets. In fact it appears $(k+1)!$ since any rearrangement of the first $k+1$ b_i's gives a permutation that has $b_1, b_2, \ldots$ in its set. Thus the number of things in all the sets is $N_k(k+1)!$. Consequently, $N_k = n!k/(k+1)!$.

By the previous paragraphs, the average number of times the "`If`" is executed is $\sum k^2/(k+1)!$, which approaches some constant. Thus the average running time is $\Theta(1)$.

(c) The minimum running time occurs when $a_n > a_{n+1}$ and this time is $\Theta(n)$. By previous results the maximum running time is also $\Theta(n)$. Thus the average running time is $\Theta(n)$.

Section B.3

B.3.1 (a) If we have know $\chi(G)$, then we can determine if c colors are enough by checking if $c \geq \chi(G)$.

(b) We know that $0 \leq \chi(G) \leq n$ for a graph with n vertices. Ask if c colors suffice for $c = 0, 1, 2, \ldots$. The least c for which the answer is "yes" is $\chi(G)$. Thus the worst case time for finding $\chi(G)$ is at most n times the worst case time for the NP-complete problem. Hence one time is O of a polynomial in n if and only if the other is.

B.3.2 (a) Since the largest possible value for K is $|S|$, one can use the idea that was used in the previous exercise for $\chi(G)$.

(b) Since K bins can hold at most KB and they must hold $\sum s$, we have $K(S, B)B \geq \sum s$.

(c) Let $FF(S, B)$ be the number of bins actually used. Each item in S must be placed in a bin and we must look into at most $FF(S, B)$ bins to do so. Assuming we associate with each bin the amount of space remaining (or used) in the bin, each look takes a constant amount of time. Thus the worst case running time is $O(FF(S, B)|S|)$. It is actually possible to arrange for about that many looks to be required: Let the first $FF(S, B) - 1$ items is S have size B and the rest have size ϵ, a very small number. The number of looks required is $1 + 2 + \cdots + (FF(S, B - 1)) + (|S| - FF(S, B) + 1)FF(S, B)$, which is $\Theta(FF(S, B)|S|)$. The analysis of average time is beyond our scope.

(d) We can never have more than one bin that is less than half full. Suppose we think we have used the algorithm correctly and that bins i and $j > i$ are at most half full. Since everything that fit into bin j could have been placed into bin i, the algorithm should have placed them there or in some earlier bin. Thus we could not have ended with more than one bin at most half full. It follows that $FF(S, B) - 1$ of the bins each contains items summing to over $B/2$. Thus $\sum s$ exceeds $(FF(S, B) - 1)B/2$ and so an earlier result

$$FF(S, B) < 1 + 2\sum s/B < 1 + 2K(S, B).$$