Before beginning this homework, please review the policies regarding group-based homework, online submission of homework, and academic integrity, all of which are available on the syllabus.

1. The longest common substring of two words $w_1 \ldots w_n$ and $v_1 \ldots v_m$ is the longest word that appears from left to right (in not necessarily consecutive positions) in the two strings. For example, ALGEBRA and GEOMETRY have longest common substring GER. Here's a recursive algorithm that finds the length of the longest common substring: $\text{LCS}(w_1 \ldots w_n, v_1 \ldots v_m)$.

(a) IF $n = 0$ return 0
(b) IF $m = 0$ return 0
(c) IF $w_1 = v_1$ return $1 + \text{LCS}(w_2 \ldots w_n, v_2 \ldots v_m)$
(d) Return $\max(\text{LCS}(w_2 \ldots w_n, v_1 \ldots v_m), \text{LCS}(w_1 \ldots w_n, v_2 \ldots v_m))$

Write a recurrence for $R(n, m)$, the number of base case calls (with one or the other string being empty) of the recursive algorithm when $w_1 \ldots w_n$ and $v_1 \ldots v_m$ have no common elements. Then show that the solution to the recurrence is $(\binom{m+n}{n})$. (10 points)

If $n = 0$ or $m = 0$, the main call is a base case, and no other is. So $R(0, m) = R(n, 0) = 1$. In all other cases, since $v$ and $w$ are disjoint, we always make one recursive call to a $w$ of size $n - 1$ and $v$ of size $m$, and one where $w$ has size $n$ and $v$ has size $m - 1$. Thus, for $n, m > 0$, $R(n, m) = R(n - 1, m) + R(n, m - 1)$.

We'll prove by induction on $k$ that if $k = n + m$, then $R(n, m) = \binom{n+m}{n}$.

If $k = 0$, and $n + m = k = 0$, then $n = m = 0$. Since $R(0, 0) = 1 = \binom{0}{0}$, the base case holds.

Assume for some $k$ that whenever $n + m = k$, $R(n, m) = \binom{n+m}{n}$. Let $n + m = k + 1$. If $n = 0$, $R(0, m) = 1 = \binom{m}{0}$, and if $m = 0$, $R(n, 0) = 1 = \binom{n}{n}$. Otherwise, both $n, m > 0$, and by the above recurrence, $R(n, m) = R(n - 1, m) + R(n, m - 1)$. Since $n + (m - 1) = k = n - 1 + m$, we can apply the induction hypothesis to both of the terms in this recurrence. So $R(n, m) = \binom{n-1+m}{n-1} + \binom{n+m-1}{n} = \binom{n+m}{n}$, where the last step is Pascal’s identity. Thus, by induction, the number of recurrences is $R(n, m) = \binom{n+m}{n}$ for any value of $n + m$.

2. Consider a process where a particle is moving in an infinite grid of pairs of non-negative integers, $(x, y)$, where it starts at $(0, 0)$ and each step it either moves up (increasing $y$ by 1) or right (increasing $x$ by 1). How many ways are there for the particle to reach the point $(n, m)$? (5 points). If each step, the particle is equally likely to move up or right, what is the probability that it ever passes through the point $(n, m)$? (5 points)

We can describe a path from the start to the point $x, y$ as a series of right moves and up moves. The total number of right moves will be $x$ and the number of up moves will be $y$. So the number of paths is equal to the number of binary strings with $x$ ones and $y$ zeroes, $\binom{x+y}{x}$. For each such path, the probability of following that path is $2^{-(x+y)}$, because $x + y$ fair coins must match the moves of the path. Since each path is a disjoint event, (you cannot follow multiple paths), the total probability is then $\binom{x+y}{x}2^{-(x+y)}$. 
3. (a) (3 points) You are dealt a five card poker hand face down (and haven’t looked at it.) What is the probability that the hand is a flush (all the same suit)?

   We know the number of hands that are flushes is \( 4 \binom{13}{5} \) from last assignment. Out of all \( \binom{52}{5} \) hands, this means the probability is
   \[
   \frac{4 \binom{13}{5}}{\binom{52}{5}}.
   \]

(b) (3 points) Then you flip over the first card and it is a heart. Is the new conditional probability that the hand is a flush larger, smaller or the same as the original probability?

   It is the same. The situation is symmetric with any other outcome (spades, clubs, diamonds). So the conditional probability cannot have changed.

(c) (4 points) You flip over the second card and it is also a heart. What is the new conditional probability that the hand is a flush?

   Since for all flushes, the second card must match the suit of the first, the over all probability is
   \[
   \text{Prob} \{\text{flush} \} = \text{Prob} \{\text{second card matches first}\} \text{Prob} \{\text{flush|second card matches first}\}
   \]

   The probability that the second card matches the first is \( \frac{12}{51} = \frac{4}{17} \), since there are 12 matching cards out of the 51 left in the deck. Thus, the new conditional probability is \( \frac{17}{4} = 4.25 \) times the original probability, which is considerably larger.

4. **10 points** Say you have a fair coin, and want to use it to simulate a biased coin with probability \( 0 < p < 1 \) of heads. Describe a method to do so exactly where the expected number of fair coin flips you use is a fixed number independent of \( p \). You need to prove that your method outputs heads with probability \( p \), and bound the expected number of coin flips it makes. (Hint: look at the expansion of \( p \) in binary, beyond the decimal point.)

   Let \( p = .p_1p_2...p_i... \) be the binary representation of \( p \). We will flip our fair coin, interpreting heads as 1, and tails as 0, to get a sequence \(.b_1b_2...b_i...\). However, as soon as \( b_i \neq p_i \), we stop, and output heads if \( b_i < p_i \) and tails otherwise. We can think of this as generating a random number from 0 to 1, and testing whether that number is less than \( p \). Or we can view it as follows: We can output heads after \( i \) flips only if \( p_i = 1 \). For each such \( i \), we output heads at the \( i \)’th step, if and only if the sequence matches \( p \) for \( i-1 \) steps and then \( b_i = 0 \). This happens with probability \( 2^{-i} \). Thus, the overall probability of outputting heads is \( \sum_{i \geq 1} 2^{-i} p_i \) (since if \( p_i = 0 \) we contribute 0 to the overall probability). But by definition of binary representation, this is \( p \).

   The expected number of coin flips that this method uses is 2, since this is a special case of the solitaire game problem from lecture with probability 1/2 of winning a game. Here, winning a game represents not matching the bit \( p_i \).

5. (10 points) In an unsorted array, an element is a **local maximum** if it is larger than both of the two adjacent elements. The first and last elements of the array are considered local maxima if they are larger than the only adjacent element. If we create an array by randomly permuting the numbers from 1 to \( n \), what is the expected number of local maxima? Prove your answer correct using additivity of expectations.

   Let \( X_i = 1 \) if \( a_i \) is a local max, and 0 otherwise. We want to bound \( E[S] \) where \( S = \sum_{i=1}^{n} X_i \). By linearity of expectations, \( E[S] = \sum E[X_i] = \sum \text{Prob}[X_i = 1] \). For \( i \neq 1, i \neq n \), there are two neighbors of \( a_i \), and all three are equally likely to be the largest of the 3. So the probability that \( a_i \) is a local max is 1/3. Similarly, since \( a_1 \) and \( a_n \) each have one neighbor, their probability of being a local max is 1/2 each. Thus, the overall expectation of \( S \) is \( 2(1/2) + (n - 2)(1/3) = (n + 1)/3 \)