## Content

<table>
<thead>
<tr>
<th>1</th>
<th>Understanding and analyzing algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Using graphs and graph algorithms</td>
</tr>
<tr>
<td>3</td>
<td>Using combinatorial reasoning and probability to quantitatively analyze algorithms and systems</td>
</tr>
</tbody>
</table>

#### 3.1 Basics of Counting: Product rule, Sum rule
- 3.2 Permutations, Combinations
- 3.3 Lower bound for comparison-based sorting
- 3.5 Probability and Counting
- 3.6 Conditional Probabilities
- 3.7 Birthday Paradox, Hashing and Randomized Algorithms
Basics of Counting: Product Rule and Sum Rule
What do we mean by counting?

- How many arrangements or combinations of objects are there of a given form?
- How many of these have a certain property?

Math = “search for order”
Counting is important for Computer Scientists

- For computer scientists:
  - **Hardware**: How many ways are there to arrange components on a chip?
  - **Algorithms**: How long is this loop going to take? How many times does it run?
  - **Security**: How many passwords are there?
  - **Memory**: How many bits of memory should be allocated to store an object?
Memory requirements

How many bits of memory should be allocated to store a decimal numbers with \( n \) digits?
Memory requirements

How many bits of memory should be allocated to store a decimal numbers with $n$ digits?

<table>
<thead>
<tr>
<th>$n$</th>
<th>#Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
</tr>
</tbody>
</table>

$$\lceil \log_2(10^n) \rceil = \lceil n \cdot \log_2(10) \rceil \approx \lceil 3.32n \rceil$$

Moral:
If we care about storage requirements, we must care about counting.
Say you are designing a video game where each player can choose a preset character or make their own character with custom facial features.

How many different characters can be created?

Say you can only choose from these 12 hair styles, and these 8 hair colors.

How many different characters can you create?
Say you are designing a video game where each player can choose a preset character or make their own character with custom facial features.

How many different characters can be created?

Say you can only choose from these 12 hair styles, and these 8 hair colors.

How many different characters can you create?

Answer: 12 x 8 = 96
Product Rule: For any sets A and B, \(|A \times B| = |A| \times |B|\).

- In our example,
  
  \[A = \{\text{hair styles}\}, \ |A| = 12\]
  
  \[B = \{\text{hair colors}\}, \ |B| = 8\]

- Specifying a character means giving an ordered pair (hair style, hair color).

- The Product Rule says the number of such pairs is
  
  \[12 \times 8 = 96\].
• Say you can choose one of the 96 custom characters or use a preset character.
• If there are 10 preset characters you can choose from, how many different characters can you be?
Sum Rule (2/3)

• Say you can choose one of the 96 custom characters or use a preset character.
• If there are 10 preset characters you can choose from, how many different characters can you be?

Answer: 96 + 10 = 106
Sum Rule (3/3)

Sum Rule: For any disjoint sets $A$ and $B$, 
$$|A \cup B| = |A| + |B|.$$ 

- In our example, 
  \[ A = \{\text{custom characters}\}, \ |A| = 96 \]
  \[ B = \{\text{preset characters}\}, \ |B| = 10 \]
- These sets are disjoint since there is no overlap.
- The Sum Rule says the total number of characters is 
  \[ 96 + 10 = 106. \]
Sum Rule: Disjointness is necessary

Sum Rule: For any disjoint sets A and B,

\[ |A \cup B| = |A| + |B|. \]

- Disjointness is necessary, otherwise the formula is not true.
Sum Rule: Inclusion/Exclusion for two sets

\[ |A \cup B| = |A| + |B| - |A \cap B| \]

# people who know Java OR C
= # people who know Java
+ # people who know C
- # people who know Java AND C

People who know Java

People who know C
Sum Rule: Inclusion/Exclusion for three sets

\[ |A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C| \]
Example Questions

Favorite Sports:

<table>
<thead>
<tr>
<th></th>
<th>Football</th>
<th>Baseball</th>
<th>Soccer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brian</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Carl</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Christine</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Jianhan</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Louise</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Ron</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Illustrate the Sum Rule together with Inclusion/Exclusion for the given table.
Example Questions

Favorite Sports:

<table>
<thead>
<tr>
<th></th>
<th>Football</th>
<th>Baseball</th>
<th>Soccer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brian</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Carl</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Christine</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Jianhan</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Louise</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Ron</td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Illustrate the Sum Rule together with Inclusion/Exclusion for the given table.

Solution: There are 3 (Football), 4 (Basketball), 4 (Soccer) X, so in total 11 X. But there are only 6 people. The computation goes as follows:

11 – (Football and Baseball) – (Football and Soccer) – (Soccer and Baseball) + (Football and Baseball and Soccer) =

11 – 2 -2 -2 + 1 = 6
Example Questions

The chairs of an auditorium are to labeled with one of the 26 uppercase English letters followed by a positive integer not exceeding 100. What is the largest number of chairs that can be labeled differently?
Example Questions

The chairs of an auditorium are to labeled with one of the 26 uppercase English letters followed by a positive integer not exceeding 100. What is the largest number of chairs that can be labeled differently?

Solution:
26x100 = 2,600 different ways that a chair can be labeled.
Example Questions

• **Example:** At an ice cream parlor, you can have your ice cream in a bowl, cake cone, or sugar cone. There are 20 different flavors available. How many single-scoop creations?
Example Questions

**Example:** At an ice cream parlor, you can have your ice cream in a bowl, cake cone, or sugar cone. There are 20 different flavors available. How many single-scoop creations?

**Solution:**
- 3 choices for your vehicle
- 20 choices for your flavor (independent of the vehicle you chose)
- So $3 \times 20 = 60$ creations
Example Questions

- **Example:** At a different ice cream parlor, the only flavors are vanilla, chocolate, and strawberry. You can order a waffle cone or a sundae. Sundaes come with your choice of caramel or hot fudge. Whipped cream and a cherry are optional.

- Note that the number of toppings depends on whether you choose a cone or a sundae.

- Break into disjoint cases:
  - 1) Cone [3 choices]
  - 2) Sundae [s choices]

- Number of desserts = 3 + s
Example Questions

- **Example:** At a different ice cream parlor, the only flavors are vanilla, chocolate, and strawberry. You can order a waffle cone or a sundae. Sundaes come with your choice of caramel or hot fudge. Whipped cream and a cherry are optional.

- Note that the number of toppings *depends on* whether you choose a cone or a sundae.

- Break into disjoint cases:
  - 1) Cone [3 choices]
  - 2) Sundae [s choices]

- Number of desserts = 3 + s

- A sundae is a sequence or a tuple:
  (vanilla/chocolate/strawberry, caramel/hot fudge, whipped cream/none, cherry/none)

- Choices in each category are completely independent

- Number of sundaes:
  \[ s = 3 \times 2 \times 2 \times 2 = 24 \]
Example Questions

• **Example:** At a different ice cream parlor, the only flavors are vanilla, chocolate, and strawberry. You can order a waffle cone or a sundae. Sundaes come with your choice of caramel or hot fudge. Whipped cream and a cherry are optional.

• Note that the number of toppings *depends on* whether you choose a cone or a sundae.

• Break into disjoint cases:
  - 1) Cone [3 choices]
  - 2) Sundae [s choices]

• Number of desserts = 3 + s

• A sundae is a sequence or a tuple:
  (vanilla/chocolate/strawberry, caramel/hot fudge, whipped cream/none, cherry/none)

• Choices in each category are completely independent

• Number of sundaes:

  \[ s = 3 \times 2 \times 2 \times 2 = 24 \]

• In total there are

  \[ 3 + s = 3 + 24 = 27 \]

dessert options.
Example Questions

Each user in a computer system has a password, which is six to eight characters long, where each character is an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?
Example Questions

Each user in a computer system has a password, which is six to eight characters long, where each character is an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

Solution:
Let $P$ be the total number of possible passwords, and let $P_6$, $P_7$, and $P_8$ denote the number of possible passwords of length 6, 7, and 8, respectively. $P = P_6 + P_7 + P_8$. 
Example Questions

Each user in a computer system has a password, which is six to eight characters long, where each character is an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

Solution:

Let $P$ be the total number of possible passwords, and let $P_6$, $P_7$, and $P_8$ denote the number of possible passwords of length 6, 7, and 8, respectively.

$$P = P_6 + P_7 + P_8.$$ 

Number of strings of six characters:

Number of strings with no digits:
Example Questions

Each user in a computer system has a password, which is six to eight characters long, where each character is an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

Solution:
Let \( P \) be the total number of possible passwords, and let \( P_6, P_7, \) and \( P_8 \) denote the number of possible passwords of length 6, 7, and 8, respectively.

\[ P = P_6 + P_7 + P_8. \]

Number of strings of six characters: \((26+10)^6\)
Number of strings with no digits: \(26^6\)
Example Questions

Each user in a computer system has a password, which is six to eight characters long, where each character is an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

Solution:
Let P be the total number of possible passwords, and let P₆, P₇, and P₈ denote the number of possible passwords of length 6, 7, and 8, respectively. 
P = P₆ + P₇ + P₈.

Number of strings of six characters: \((26+10)^6\)
Number of strings with no digits: \(26^6\)
Hence, \(P₆ = 36^6 - 26^6 = 1.867.866.560\)

Similarly, \(P₇ = 36^7 - 26^7 = 70.332.353.920\)
and \(P₈ = 36^8 - 26^8 = 2.612.282.842.880\)

Consequently, \(P = P₆ + P₇ + P₈ = 2.684.483.063.360\).
A computer company receives 350 applications from graduates. 220 of these applicants majored in computer science, 147 in business, and 51 majored both in computer science and in business. How many of the applicants majored neither in computer science nor in business?
Example Questions

A computer company receives 350 applications from graduates. 220 of these applicants majored in computer science, 147 in business, and 51 majored both in computer science and in business. How many of the applicants majored neither in computer science nor in business?

Solution:

\[ 350 - (220 + 147 - 51) = \]
\[ 350 - 316 = \]
\[ 34 \]

34 of the applicants majored neither in computer science nor in business.
<table>
<thead>
<tr>
<th>1</th>
<th>Understanding and analyzing algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Using graphs and graph algorithms</td>
</tr>
<tr>
<td>3</td>
<td>Using combinatorial reasoning and probability to quantitatively analyze algorithms and systems</td>
</tr>
<tr>
<td>3.1</td>
<td>Basics of Counting: Product rule, Sum rule</td>
</tr>
<tr>
<td>3.2</td>
<td>Permutations, Combinations</td>
</tr>
<tr>
<td>3.3</td>
<td>Lower bound for comparison-based sorting</td>
</tr>
<tr>
<td>3.4</td>
<td>Probability - Basics: Probability in Computer Science, Experiments, Sample Space, Events, Random Variables, Expected Value, Probability Distributions (Uniform and Binomial Distribution)</td>
</tr>
<tr>
<td>3.5</td>
<td>Probability and Counting</td>
</tr>
<tr>
<td>3.6</td>
<td>Conditional Probabilities</td>
</tr>
<tr>
<td>3.7</td>
<td>Birthday Paradox, Hashing and Randomized Algorithms</td>
</tr>
</tbody>
</table>
Permutations

Combinations
Binomial Coefficients and Identities

Let \( n \) and \( k \) be nonnegative integers with \( k \leq n \). Then

1. \( C(n, k) = \binom{n}{k} = \frac{n!}{k!(n-k)!} \) (Binomial Coefficient)

2. \( \sum_{k=0}^{n} \binom{n}{k} = 2^n \)

3. \( \sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0 \)

4. \( \sum_{k=0}^{n} 2^k \binom{n}{k} = 3^n \)

5. \( \binom{n+1}{k} = \binom{n}{k-1} \binom{n}{k} \) (Pascal’s Identity)

6. \( \sum_{k=0}^{n} \binom{n}{k}^2 = \binom{2n}{n} \)
Permutations and Combinations

<table>
<thead>
<tr>
<th>No repetition of objects</th>
<th>When some of the objects are not distinguishable</th>
<th>With repetition of objects</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Permutation</strong></td>
<td><strong>Combination</strong></td>
<td><strong>C Yes/No/Maybe</strong></td>
</tr>
<tr>
<td>Ordered arrangement of ( k ) out of ( n ) objects</td>
<td>Unordered selection of ( k ) out of ( n ) objects</td>
<td>( n^k )</td>
</tr>
<tr>
<td>( \frac{n!}{(n-k)!} )</td>
<td>( \binom{n}{k} )</td>
<td>( \binom{n+k-1}{k} )</td>
</tr>
</tbody>
</table>

*Permutation*:
- Ordered arrangement
- \( \frac{n!}{(n-k)!} \)

*Combination*:
- Unordered selection
- \( \binom{n}{k} \)

**A 100m-Sprint**
- \( \frac{n!}{(n-k)!} \)

**B Necklace**
- \( \frac{n!}{n_1! \ldots n_k!} \)

**D Lottery**
- \( \binom{n}{k} \)

**E Dice**
- \( \binom{n+k-1}{k} \)
Permutation

Ordered arrangement of \( k \) out of \( n \) objects

\[
\text{A 100m-Sprint} \quad \frac{n!}{(n-k)!}
\]

\[
\text{B Necklace} \quad \frac{n!}{n_1! \ldots n_k!}
\]

\[
\text{C Yes/No/Maybe} \quad n^k
\]

### A 100m-Sprint

- \( n=3 \) sprinters A, B, C.
- Possible outcomes for the first \( k=2 \) places (no ties).

**Objects:**
- Sprinters: A, B, C (first place), A, B, C (second place)

\[
\begin{align*}
\text{AB} & \quad \text{BA} & \quad \text{CA} \\
\text{AC} & \quad \text{BC} & \quad \text{--}
\end{align*}
\]

### B Necklace

- \( n=3 \) beads red, red, green.
- Possible different patterns when you thread \( n_1=2 \) red, \( n_2=1 \) green beads on a necklace.

**Objects:**
- Beads: r, r, g (first position), r, r, g (second position), r, r, g (third position)

\[
\begin{align*}
\text{rgb} & \quad \text{brg} & \quad \text{grb} \\
\text{rgb} & \quad \text{bgr} & \quad \text{grr} \\
\text{rrg} & \quad \text{rgr} & \quad \text{grr}
\end{align*}
\]

### C Yes/No/Maybe

- \( n=3 \) answers yes, no, maybe.
- Possible outcomes for \( k=2 \) questions.

**Objects:**
- Answers: y, n, m (first question), y, n, m (second question)

\[
\begin{align*}
\text{yy} & \quad \text{ny} & \quad \text{my} \\
\text{yn} & \quad \text{nn} & \quad \text{mn} \\
\text{ym} & \quad \text{nm} & \quad \text{mm}
\end{align*}
\]
Combination

**Combination**

Unordered selection of \( k \) out of \( n \) objects

\[
\binom{n}{k}
\]

**D Lottery**

\( n = 5 \) numbers 1, 2, 3, 4, 5.
Possible outcomes for drawing \( k = 2 \) out of the \( n = 5 \) numbers.

Objects:
Numbers
1, 2, 3, 4, 5 (first number)
1, 2, 3, 4, 5 (second number)

\[
\begin{array}{cccc}
\text{12} & \text{--} \\
\text{13} & \text{23} & \text{--} \\
\text{14} & \text{24} & \text{34} & \text{--} \\
\text{15} & \text{25} & \text{35} & \text{45} & \text{--} \\
\end{array}
\]

**E Dice**

\( n = 6 \) numbers 1, 2, 3, 4, 5, 6.
Possible outcomes for \( k = 2 \) dice.

Objects:
Numbers
1, 2, 3, 4, 5, 6 (first die)
1, 2, 3, 4, 5, 6 (second die)

11
12
13
14
15
16
12
22
33
44
55
66
Example Questions

How many different 3-digit numbers can you form out of the digits 1, 2, 3?
Each digit can occur only once in a number.
Hint: the objects are 1, 2, 3
Example Questions

How many different 3-digit numbers can you form out of the digits 1, 2, 3?
Each digit can occur only once in a number.
Hint: the objects are 1, 2, 3

Solution:
• Order is important → permutation
• no repetition of objects
• all objects are distinguishable → k-permutation (k=n=3)
Example Questions

How many different 3-digit numbers can you form out of the digits 1, 2, 3?
Each digit can occur only once in a number.
Hint: the objects are 1, 2, 3

Solution:
• Order is important → permutation
• no repetition of objects
• all objects are distinguishable → k-permutation (k=n=3)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>213</td>
<td>312</td>
</tr>
<tr>
<td>132</td>
<td>231</td>
<td>321</td>
</tr>
</tbody>
</table>

\[
n! \over (n-k)!\]

\(n=3\) sprinters A, B, C. Possible outcomes for the first \(k=2\) places (no ties).

Objects:
Sprinters
A, B, C (first place)
A, B, C (second place)

\|--|--|--\|
|BA|CA|\|--|--|--\|
|AB|CB|\|--|--|--\|
|AC|BC|--|
Example Questions

How many different committees of three students can be formed from a group of four students?
Hint: the objects (students) are numbered 1, 2, 3, 4
Example Questions

How many different committees of three students can be formed from a group of four students?
Hint: the objects (students) are numbered 1, 2, 3, 4

Solution:
• Order doesn’t matter → combination
• no repetition of objects → k-combination (k=3, n=4)
Example Questions

How many different committees of three students can be formed from a group of four students?
Hint: the objects (students) are numbered 1, 2, 3, 4

Solution:
• Order doesn’t matter → combination
• no repetition of objects → k-combination (k=3, n=4)

123
124
134
234

D Lottery
\[
\binom{n}{k}
\]

n=5 numbers 1, 2, 3, 4, 5. Possible outcomes for drawing k=2 out of the n=5 numbers.

Objects:
Numbers
1, 2, 3, 4, 5 (first number)
1, 2, 3, 4, 5 (second number)

<table>
<thead>
<tr>
<th></th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>13</td>
<td>23</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>14</td>
<td>24</td>
<td>34</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>15</td>
<td>25</td>
<td>35</td>
<td>45</td>
<td>--</td>
</tr>
</tbody>
</table>
Example Questions

Suppose a department contains 10 men and 15 women. How many ways are there to form a committee with six members if it must have the same number of men and women?
Example Questions

Suppose a department contains 10 men and 15 women. How many ways are there to form a committee with six members if it must have the same number of men and women?

Solution:
• We have to choose 3 out of 10 men and 3 out of 15 women.
• Order doesn’t matter → combination
• no repetition of objects → k-combination
men: k=3, n=10
women: k=3, n=15

\[
\binom{10}{3} \cdot \binom{15}{3} = \frac{10 \cdot 9 \cdot 8}{3 \cdot 2} \cdot \frac{15 \cdot 14 \cdot 13}{3 \cdot 2} = 120 \cdot 455 = 54,600
\]
More Examples

Homework and Rosen, Chapter 6.
<table>
<thead>
<tr>
<th>1</th>
<th>Understanding and analyzing algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Using graphs and graph algorithms</td>
</tr>
<tr>
<td>3</td>
<td>Using combinatorial reasoning and probability to quantitatively analyze algorithms and systems</td>
</tr>
</tbody>
</table>

| 3.1 Basics of Counting: Product rule, Sum rule |
| 3.2 Permutations, Combinations |
| 3.3 **Lower bound for comparison-based sorting** |
| 3.4 Probability - Basics: Probability in Computer Science, Experiments, Sample Space, Events, Random Variables, Expected Value, Probability Distributions (Uniform and Binomial Distribution) |
| 3.5 Probability and Counting |
| 3.6 Conditional Probabilities |
| 3.7 Birthday Paradox, Hashing and Randomized Algorithms |
Lower bound for comparison-based sorting
Lower bound for comparison-based sorting

We measure the cost of a sorting algorithm in the number of comparisons between array elements.

We will show in this section that best comparison-based sorting algorithms are \( \Theta(n \log(n)) \)
(like MergeSort)

So it is impossible to have a comparison-based algorithm that does better than this in the worst case.
We can construct a *branching diagram* or *decision tree* that shows the possible comparisons we might have to do.

**Binary Tree**

**Internal vertices:** Comparisons

**Leaves:** All possible sorted orders (permutations) of the array of size $n$

**Question 1:** How many leaves are there?
Decision Tree

We can construct a *branching diagram* or *decision tree* that shows the possible comparisons we might have to do.

**Binary Tree**

**Internal vertices:**
Comparisons

**Leaves:**
All possible sorted orders (permutations) of the array of size $n$

**Question 1:** How many leaves are there?

$n!$

---

CSE 21
SS2, 2017
Prof. Dr. Oliver Braun

1 Analyzing Algorithms | 2 Graphs and graph algorithms | 3 Combinatorial reasoning and Probability

3.1 Counting | 3.2 Perm/Comb | **3.3 Sorting** | 3.4 Prob.-Basics | 3.5 Prob. and Counting | 3.6 Contitional Prob. | 3.7 Randomized Alg.
We can construct a *branching diagram* or *decision tree* that shows the possible comparisons we might have to do.

**Binary Tree**

**Internal vertices:**
Comparisons

**Leaves:**
All possible sorted orders (permutations) of the array of size $n$

**Question 2:** The maximum number of comparisons we might have to make (worst case) is the height of the tree. What is the height of a decision tree that is used to sort an array of size $n$?
Decision Tree

We can construct a *branching diagram* or *decision tree* that shows the possible comparisons we might have to do.

![Decision Tree Diagram]

**Binary Tree**

**Internal vertices:** Comparisons

**Leaves:** All possible sorted orders (permutations) of the array of size $n$

**Question 2:** *The maximum number of comparisons we might have to make (worst case) is the height of the tree. What is the height of a decision tree that is used to sort an array of size $n$?*

$log(n!)$
Height of the Decision Tree

If a binary tree has height = $k$, then it has $\leq 2^k$ leaves.
If a binary tree has height $< k$, then it has $< 2^k$ leaves.
If a binary tree has height $< \log(k)$, then it has $< k$ leaves.
Height of the Decision Tree

If a binary tree has height $= k$, then it has $\leq 2^k$ leaves.
If a binary tree has height $< k$, then it has $< 2^k$ leaves.
If a binary tree has height $< \log(k)$, then it has $< k$ leaves.

If a binary tree has $\geq k$ leaves, then it has has height $\geq \log(k)$. $\leftarrow$ Contrapositive
If a binary tree has $\geq n!$ leaves, then it has has height $\geq \log(n!)$. 
If a binary tree has height $= k$, then it has $\leq 2^k$ leaves.
If a binary tree has height $< k$, then it has $< 2^k$ leaves.
If a binary tree has height $< \log(k)$, then it has $< k$ leaves.

If a binary tree has $\geq k$ leaves, then it has has height $\geq \log(k)$. ← Contrapositive
If a binary tree has $\geq n!$ leaves, then it has has height $\geq \log(n!)$.

This says the branching diagram for any sorting algorithm has height $\geq \log(n!)$.

Since the number of comparisons was the height of the tree, the worst case for any sorting algorithm is $\geq \log(n!)$ comparisons.
How big is $\log(n!)$?

Lemma 1: For $n > 1$, $(\frac{n}{2})^{\frac{n}{2}} < n! < n^n$.

Proof:

$$n! = n \cdot (n - 1) \cdot (n - 2) \cdot \ldots \cdot \frac{n}{2} \cdot \ldots \cdot 3 \cdot 2 \cdot 1$$

$$> \frac{n}{2} \cdot \frac{n}{2} \cdot \frac{n}{2} \cdot \ldots \cdot \frac{n}{2}$$

$$= \left(\frac{n}{2}\right)^{\frac{n}{2}}$$

$$n! = n \cdot (n - 1) \cdot (n - 2) \cdot \ldots \cdot 3 \cdot 2 \cdot 1$$

$$< n \cdot n \cdot n \cdot \ldots \cdot n \cdot n \cdot n$$

$$= (n^n)$$
How big is log(n!)?

Lemma 2: \(\log(n!)\) is in \(\Theta(n \log n)\).

Proof:

\[
\left( \frac{n}{2} \right)^{\frac{n}{2}} < n! < n^n
\]

\[
\log \left( \frac{n}{2} \right)^{\frac{n}{2}} < \log(n!) < \log(n^n)
\]

\[
\frac{n}{2} \log \left( \frac{n}{2} \right) < \log(n!) < n \log(n)
\]
Lower bound for comparison-based sorting

Result:

The best comparison-based sorting algorithms are \(\Theta(n \log(n))\) like MergeSort.

It is impossible to have a comparison-based algorithm that does better than this in the worst case.
## Content

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Understanding and analyzing algorithms</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Using graphs and graph algorithms</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Using combinatorial reasoning and probability to quantitatively analyze algorithms and systems</td>
<td>3.1 Basics of Counting: Product rule, Sum rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.2 Permutations, Combinations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.3 Lower bound for comparison-based sorting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.4 Probability - Basics: Probability in Computer Science, Experiments, Sample Space, Events, Random Variables, Expected Value, Probability Distributions (Uniform and Binomial Distribution)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.5 Probability and Counting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.6 Conditional Probabilities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.7 Birthday Paradox, Hashing and Randomized Algorithms</td>
</tr>
</tbody>
</table>
Basics
Probability is important in computer science, because sometimes

- the input is random
  (Data Mining, analyzing data from experiments),
- the desired output is random
  (picking a cryptographic key, performing a scientific simulation),
- the algorithm uses randomness
  (Monte Carlo methods, Search Heuristics, Randomized Hashing, Quicksort).
Probability in Computer Science

Probability is important in computer science, because sometimes

- the input is random (Data Mining, analyzing data from experiments),
- the desired output is random (picking a cryptographic key, performing a scientific simulation),
- the algorithm uses randomness (Monte Carlo methods, Search Heuristics, Randomized Hashing, Quicksort).

In Data Mining and Machine Learning, algorithms are used to understand information and make predictions.

- These can be for elections, the weather, stock prices, or genetic indicators for diseases.
- Much of that **BIG DATA** information comes from some kind of randomized sampling: polls, statistics, random probes.
- An understanding of probability is needed to distinguish between valid predictions and overfitting to particular data.
Simulations

In simulations, you want to examine “typical” behaviour of a system, so you want to look at random events conditioned on a complex set of constraints. For example, the chart below shows the results of randomized stock price simulations, performed many times.
Randomized Algorithms

Sometimes randomness can be used by algorithms even when the answer we want is deterministic:

- For example, suppose you are trying to find the minimum possible value of a function.
- In the metropolis heuristic, random moves are combined with greedy moves to favor smaller values without getting stuck at local minima.
Basic Definitions and Examples

Experiments
- Toss a coin
- Roll a die
- Penalty
- Shootout
- Sunshine
today

Sample Space and Events
A subset $E$ of $S$ is called an event, and we define the probability of $E$, by $\text{Prob}[E] = \sum_{s \in E} p_s$.

Random Variables
Formally, a random variable $X = F(s)$ is determined by a function $F : S \rightarrow \mathbb{R}$. The distribution for $X$ is $p_v = \text{Prob}[X = v]$ for each possible value $v$.

Probability Distributions
A probability distribution is an assignment of probabilities $0 \leq p_s \leq 1$ to each element of a sample space $S$ so that

$$\sum_{s \in S} p_s = 1.$$
Basic Definitions and Examples

Experiments | Toss a coin | Roll a die | Penalty Shootout | Sunshine today

Sample Space and Events

Random Variables

Probability

Distributions
For example, consider the sample space of all possible outcomes of a six-sided die, and let each have probability 1/6. We identify the event “the die roll is even,” with the set of outcomes where that is true:

\{2, 4, 6\}

and the total probability of that event,

\[ \text{Prob}[\text{Even}] = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}. \]
Uniform and Binomial Distribution

- A particularly natural type of distribution is the *uniform* distribution, where all outcomes are equally likely, i.e., $p_s = 1/|S|$ for all $s \in S$.

- This distribution, where the sample space is $\{0$ heads, $1$ head, ..., $n$ heads$\}$, is called the *binomial distribution*, because the probabilities are proportional to the binomial coefficients.
**Binomial Distribution**

\[ n=5 \text{ (Shots on the goal)} \]
\[ \pi=70\% \quad \text{(Prob. for Goal)} \]
\[ (1-\pi)=30\% \quad \text{(Prob. for No-Goal)} \]

\[ P(X=5)= P(GGGGG)=0.7^5=16.81\% \]

\[ P(X=0)= P(NNNNN)=0.3^5=0.24\% \]

\[ P(X=4)= P(GGGGN)+ P(GGGNG)+ P(GGGGG)+ P(GNGGG)+ P(NGGGG) = 5 \times 0.7^4 \times 0.3 = 36.02\% \]

\[ P(X=1)= P(GNNNN)+ P(GNNGN)+ P(GNNGN)+ P(GNNGN)+ P(NGGNN)+ P(NNGGN)+ P(NNNGN)+ P(NNNGN)+ P(NNNGN)+ P(NNNGN) = 10 \times 0.7^2 \times 0.3^3 = 13.23\% \]

\[ P(X=2)= P(GGGNN)+ P(GGGNN)+ P(GGGNN)+ P(GGGNN)+ P(NGGNN)+ P(NNGGN)+ P(NNNGN)+ P(NNNGN)+ P(NNNGN)+ P(NNNGN) = 10 \times 0.7^3 \times 0.3^2 = 30.87\% \]
Expected Value of a Random Variable

We often want to look at the average or expected value of a random variable.

The expectation of $X$, written $E[X]$, is defined to be

$$E[X] = \sum_v v \cdot \text{Prob}[X = v]$$

where the sum is over all possible outcomes $v$ of the variable $X$.

<table>
<thead>
<tr>
<th>Penalty Shootout</th>
<th>Roll a die</th>
<th>Toss a coin</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0,00</td>
</tr>
<tr>
<td>2</td>
<td>3,23%</td>
<td>0,26</td>
</tr>
<tr>
<td>3</td>
<td>30,87%</td>
<td>0,93</td>
</tr>
<tr>
<td>4</td>
<td>36,02%</td>
<td>1,44</td>
</tr>
<tr>
<td>5</td>
<td>16,81%</td>
<td>0,84</td>
</tr>
<tr>
<td></td>
<td>100,01%</td>
<td>3,50 Exp.Value</td>
</tr>
<tr>
<td>0</td>
<td>16,67%</td>
<td>0,17</td>
</tr>
<tr>
<td>1</td>
<td>16,67%</td>
<td>0,33</td>
</tr>
<tr>
<td>2</td>
<td>16,67%</td>
<td>0,50</td>
</tr>
<tr>
<td>3</td>
<td>16,67%</td>
<td>0,67</td>
</tr>
<tr>
<td>4</td>
<td>16,67%</td>
<td>0,83</td>
</tr>
<tr>
<td>5</td>
<td>16,67%</td>
<td>1,00</td>
</tr>
<tr>
<td></td>
<td>100,00%</td>
<td>3,50 Exp.Value</td>
</tr>
</tbody>
</table>

### Table 1

<table>
<thead>
<tr>
<th>Event</th>
<th>Probability</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalty</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shootout</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Roll a die</td>
<td>16,67%</td>
<td>0,17</td>
</tr>
<tr>
<td>Toss a coin</td>
<td>0</td>
<td>0,00</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0,17</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0,33</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0,50</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0,67</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0,83</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1,00</td>
</tr>
<tr>
<td></td>
<td>100,00%</td>
<td>3,50 Exp.Value</td>
</tr>
</tbody>
</table>
Example

20% of all notebooks must be repaired during the first two years.
What is the probability that out of 5 notebooks
a) 0
b) exactly 1
c) 1 or more
have to be repaired in the first two years?
Hypothesis testing

When hypotheses are tested for statistical significance, the probability for an error that one decides that the hypothesis is true although the hypothesis is false is often set to $\alpha=5\%$. That is, in 5\% of all cases one decides that a hypothesis is true although it is false.

What is the probability that in 10 independent tests of a hypothesis this hypothesis is
a) never
b) exactly one time
c) at most two times
assumed to be true although it is false?
<table>
<thead>
<tr>
<th>1</th>
<th>Understanding and analyzing algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Using graphs and graph algorithms</td>
</tr>
<tr>
<td>3</td>
<td>Using combinatorial reasoning and probability to quantitatively analyze algorithms and systems</td>
</tr>
</tbody>
</table>

### 3.1 Basics of Counting: Product rule, Sum rule

### 3.2 Permutations, Combinations

### 3.3 Lower bound for comparison-based sorting

### 3.4 Probability - Basics: Probability in Computer Science, Experiments, Sample Space, Events, Random Variables, Expected Value, Probability Distributions (Uniform and Binomial Distribution)

#### 3.5 Probability and Counting

#### 3.6 Conditional Probabilities

#### 3.7 Birthday Paradox, Hashing and Randomized Algorithms
Probability and Counting
Computing probabilities is very connected to counting. If we start with the uniform distribution on a set $S$, and look at the probability of an event $E$ ($E \subseteq S$), then

$$\text{Prob}[E] = \frac{|E|}{|S|}.$$ 

To generalize the distribution before, if we flip $n$ fair coins (i.e., the uniform distribution on sequences of length $n$), what will the probability of getting exactly $k$ heads be? 

**Caution:** You must have each outcome equally likely to use this reasoning.
Example

Suppose 5-card hands are dealt at random from a standard deck of 52. What is the probability that your hand contains exactly two Aces?
Example

Suppose 5-card hands are dealt at random from a standard deck of 52. What is the probability that your hand contains exactly two Aces?

Solution:
C(4,2) 2 aces out of 4 possible aces
C(48,3) 3 cards out of 48 (no aces)
C(52,5) 5 cards out of 52
Example

Suppose 5-card hands are dealt at random from a standard deck of 52. What is the probability that your hand contains exactly two Aces?

Solution:

\[ \binom{4}{2} \] 2 aces out of 4 possible aces
\[ \binom{48}{3} \] 3 cards out of 48 (no aces)
\[ \binom{52}{5} \] 5 cards out of 52

\[ \frac{\binom{4}{2} \times \binom{48}{3}}{\binom{52}{5}} \]
\[ = \frac{(47 \times 46 \times 5 \times 4 \times 3 \times 2)}{(52 \times 51 \times 50 \times 49)} \]
\[ = 3.99\% \]
A rise in a permutation of the numbers \{1,...,n\} occurs when a larger number immediately follows a smaller one. For example, if \(n=5\), the permutation 1 3 2 4 5 has three rises. What is the expected number of rises in a permutation of size \(n\)?
Example

A rise in a permutation of the numbers \{1,\ldots,n\} occurs when a larger number immediately follows a smaller one. For example, if \(n=5\), the permutation 1 3 2 4 5 has three rises. What is the expected number of rises in a permutation of size \(n\)?

Solution:
The expected number of rises from position \(i\) to \(i+1\) is \(1/2\) ("linearity of expectations").
We have \(n-1\) instances of this, therefore the expected number of rises in a permutation of size \(n\) is \((1/2) \times (n-1)\).
Example

A rise in a permutation of the numbers \{1,...,n\} occurs when a larger number immediately follows a smaller one. For example, if \(n=5\), the permutation

\[1 \, 3 \, 2 \, 4 \, 5\]

has three rises. What is the expected number of rises in a permutation of size \(n\)?

Solution:
The expected number of rises from position \(i\) to \(i+1\) is \(1/2\) ("linearity of expectations").

We have \(n-1\) instances of this, therefore the expected number of rises in a permutation of size \(n\) is \((1/2) \times (n-1)\).

Example (\(n=3\)):

<table>
<thead>
<tr>
<th>Example (n=3):</th>
<th>exp.nr.of rises</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>2</td>
</tr>
<tr>
<td>(1/2)x(3-1)=1</td>
<td>1</td>
</tr>
<tr>
<td>132</td>
<td>1</td>
</tr>
<tr>
<td>213</td>
<td>1</td>
</tr>
<tr>
<td>231</td>
<td>1</td>
</tr>
<tr>
<td>312</td>
<td>1</td>
</tr>
<tr>
<td>321</td>
<td>0</td>
</tr>
<tr>
<td>Content</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>1 Understanding and analyzing algorithms</td>
<td></td>
</tr>
<tr>
<td>2 Using graphs and graph algorithms</td>
<td></td>
</tr>
</tbody>
</table>
| 3 Using combinatorial reasoning and probability to quantitatively analyze algorithms and systems | 3.1 Basics of Counting: Product rule, Sum rule  
3.2 Permutations, Combinations  
3.3 Lower bound for comparison-based sorting  
3.4 Probability - Basics: Probability in Computer Science, Experiments, Sample Space, Events, Random Variables, Expected Value, Probability Distributions (Uniform and Binomial Distribution)  
3.5 Probability and Counting  
3.6 Conditional Probabilities  
3.7 Birthday Paradox, Hashing and Randomized Algorithms |
Conditional Probabilities
Conditional Probabilities

If we initially have a certain probability distribution on outcomes, but then learn something about the actual outcome, this changes our thinking according to the Conditional Probability Law.

The conditional probability of an outcome \( s \), given an event \( E \), written \( \text{Prob}[s|E] \) is

- 0 if \( s \notin E \), and
- \( \frac{p_s}{\text{Prob}[E]} \) otherwise.

So if I roll a die, and I don’t look at it, my initial probability distribution is that each of 6 outcomes has probability 1/6. But if you tell me the result is even, my new probability distribution is that each of 2, 4 and 6 have probability 1/3.
More generally, the Conditional Probability Law says how to update probabilities of one event $B$ if we discover that another event $A$ has taken place:

**Theorem (Conditional Probability Law)**

For any events $A$, $B$,

\[ \text{Prob}[B|A] = \frac{\text{Prob}[A \land B]}{\text{Prob}[A]} . \]

Often, we use this in the other direction:

**Theorem (Conditional Probability Law)**

For any events $A$, $B$,

\[ \text{Prob}[A \land B] = \text{Prob}[A]\text{Prob}[B|A] . \]
Problem

Here's a puzzle that stumps many people, although it's just a simple calculation. Assume that initially boys and girls are equally likely.

1. Ms. X has two children. If you know the oldest is a girl:
   What is the probability that both are girls?    Answer: 50%

2. Mr. Y has two children. If you know that one of them is a boy:
   What is the probability that both are boys?    Answer: 33%

What is the sample space?
bb, bg, gb, gg

What is our initial distribution on the sample space?
Each element of the sample space has probability 1/4 (the uniform distribution).
Problem

We know Ms. X's oldest child is a girl. So if we list the children in age order, we are conditioning on the event $A = \{\text{gb, gg}\}$. The event we want the probability for is $B = \{\text{gg}\}$.

$$
Prob[B|A] = \frac{Prob[A \land B]}{Prob[A]} = \frac{1/4}{1/2} = 1/2.
$$

We know one of Mr. Y's children is a boy. So if we list the children in age order, we are conditioning on the event $A = \{\text{bb, gb, bg}\}$. The event we want the probability for is $B = \{\text{bb}\}$.

$$
Prob[B|A] = \frac{Prob[A \land B]}{Prob[A]} = \frac{1/4}{3/4} = 1/3.
$$
Example

A bitstring of length 4 is generated randomly one bit at a time. So far, you can see that the first bit is a 1. What is the probability that the string will have at least two consecutive 0's?
Example

A bitstring of length 4 is generated randomly one bit at a time. So far, you can see that the first bit is a 1. What is the probability that the string will have at least two consecutive 0's?

Solution:
We are conditioning on the event $A = \{\text{first bit is a 1}\}$.
The event we want the probability for is $B = \{\text{at least two consecutive 0's}\}$. 
Example

A bitstring of length 4 is generated randomly one bit at a time. So far, you can see that the first bit is a 1. What is the probability that the string will have at least two consecutive 0's?

Solution:
We are conditioning on the event $A=\{\text{first bit is a 1}\}$. The event we want the probability for is $B=\{\text{at least two consecutive 0’s}\}$.

$$P(A) = \frac{1}{2}$$
$$P(A \text{ and } B) = \frac{3}{16} \text{ (last 3 bits must be 000 or 001 or 100)}$$
$$P(B|A) = \frac{P(A \text{ and } B)}{P(A)} = \frac{3/16}{1/2} = \frac{3}{8}$$
Simpson’s Paradox (1/2)

In the early 1970's, UC Berkeley was sued under the Equal Opportunity Act.
The plaintiffs showed that only 35% of women who applied to graduate school were accepted in 1973, compared to 45% of men who applied.
The University countered by showing that, in every department, the percentage of women who were accepted was at least as large as the percentage of men who were accepted.
How are both of these possible at the same time?

We have three events for random applicants: Male, Female, Accepted.
We know
\[
\Pr[\text{Accepted}|\text{Male}] > \Pr[\text{Accepted}|\text{Female}].
\]
But for each department,
\[
\Pr[\text{Accepted}|\text{Male, Department}] \leq \Pr[\text{Accepted}|\text{Female, Department}].
\]
Is this possible?
Simpson's Paradox (2/2)

We can show this phenomenon is possible by giving a situation where it occurs.

<table>
<thead>
<tr>
<th>Math Department</th>
<th>English Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>60% of females accepted</td>
<td>20% of females accepted</td>
</tr>
<tr>
<td>50% of males accepted</td>
<td>10% of males accepted</td>
</tr>
</tbody>
</table>

300 females apply for Math, and 700 for English.
700 males apply for Math, and 300 for English.
## Content

<table>
<thead>
<tr>
<th>1</th>
<th>Understanding and analyzing algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Using graphs and graph algorithms</td>
</tr>
<tr>
<td>3</td>
<td>Using combinatorial reasoning and probability to quantitatively analyze algorithms and systems</td>
</tr>
</tbody>
</table>

**Part 3: Using combinatorial reasoning and probability to quantitatively analyze algorithms and systems**

- **3.1 Basics of Counting:** Product rule, Sum rule
- **3.2 Permutations, Combinations**
- **3.3 Lower bound for comparison-based sorting**
- **3.4 Probability - Basics:** Probability in Computer Science, Experiments, Sample Space, Events, Random Variables, Expected Value, Probability Distributions (Uniform and Binomial Distribution)
- **3.5 Probability and Counting**
- **3.6 Conditional Probabilities**
- **3.7 Birthday Paradox, Hashing and Randomized Algorithms**
Birthday Paradox, Hashing and Randomized Algorithms
Birthday TODAY

Find the smallest number of people you need to choose at random so that the probability that at least one of them has a birthday today exceeds 1/2.
Birthday TODAY

Find the smallest number of people you need to choose at random so that the probability that at least one of them has a birthday today exceeds 1/2.

Solution:
Assuming a year has 365 days, the probability of someone not having a birthday today is 364/365.
Given n people, the probability of them having a birthday today is (364/365)^n.
Birthday TODAY

Find the smallest number of people you need to choose at random so that the probability that at least one of them has a birthday today exceeds 1/2.

Solution:
Assuming a year has 365 days, the probability of someone not having a birthday today is 364/365.
Given n people, the probability of them having a birthday today is $(364/365)^n$.
So we need $1 - (364/365)^n$ $\geq$ 1/2, or $n$ $\geq$ $\log_{(364/365)} 1/2$, so $n$ $\geq$ 253.
Birthday Paradox

What is the minimum number of people who need to be in a room so that the probability that at least two of them have the same birthday is greater than 50%?

We make the following assumptions: The birthdays of the people in the room are independent, each birthday is equally likely, there are 366 days in the year.
Birthday Paradox

What is the minimum number of people who need to be in a room so that the probability that at least two of them have the same birthday is greater than 50%?

We make the following assumptions: The birthdays of the people in the room are independent, each birthday is equally likely, there are 366 days in the year.

\[ X = \text{“at least two out of } n \text{ people in a room have the same birthday”} \]

\[ P(X) = 1 - P(Y) \]

\[ Y = \text{“all } n \text{ people in the room have different birthdays”} \]

\[ P(Y) = \frac{365}{366} \times \frac{364}{366} \times \frac{363}{366} \times \ldots \times \frac{367-n}{366} \]
## Birthday Paradox

<table>
<thead>
<tr>
<th>k</th>
<th>p_k</th>
<th>q_k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.997</td>
<td>0.003</td>
</tr>
<tr>
<td>3</td>
<td>0.995</td>
<td>0.008</td>
</tr>
<tr>
<td>4</td>
<td>0.992</td>
<td>0.016</td>
</tr>
<tr>
<td>5</td>
<td>0.989</td>
<td>0.027</td>
</tr>
<tr>
<td>6</td>
<td>0.986</td>
<td>0.040</td>
</tr>
<tr>
<td>7</td>
<td>0.984</td>
<td>0.056</td>
</tr>
<tr>
<td>8</td>
<td>0.981</td>
<td>0.074</td>
</tr>
<tr>
<td>9</td>
<td>0.978</td>
<td>0.094</td>
</tr>
<tr>
<td>10</td>
<td>0.975</td>
<td>0.117</td>
</tr>
<tr>
<td>11</td>
<td>0.973</td>
<td>0.141</td>
</tr>
<tr>
<td>12</td>
<td>0.970</td>
<td>0.167</td>
</tr>
<tr>
<td>13</td>
<td>0.967</td>
<td>0.194</td>
</tr>
<tr>
<td>14</td>
<td>0.964</td>
<td>0.223</td>
</tr>
<tr>
<td>15</td>
<td>0.962</td>
<td>0.252</td>
</tr>
<tr>
<td>16</td>
<td>0.959</td>
<td>0.283</td>
</tr>
<tr>
<td>17</td>
<td>0.956</td>
<td>0.314</td>
</tr>
<tr>
<td>18</td>
<td>0.954</td>
<td>0.346</td>
</tr>
<tr>
<td>19</td>
<td>0.951</td>
<td>0.378</td>
</tr>
<tr>
<td>20</td>
<td>0.948</td>
<td>0.411</td>
</tr>
<tr>
<td>21</td>
<td>0.945</td>
<td>0.443</td>
</tr>
<tr>
<td>22</td>
<td>0.943</td>
<td>0.475</td>
</tr>
<tr>
<td>23</td>
<td>0.940</td>
<td>0.506</td>
</tr>
<tr>
<td>24</td>
<td>0.937</td>
<td>0.537</td>
</tr>
<tr>
<td>25</td>
<td>0.934</td>
<td>0.568</td>
</tr>
<tr>
<td>26</td>
<td>0.932</td>
<td>0.597</td>
</tr>
<tr>
<td>27</td>
<td>0.929</td>
<td>0.626</td>
</tr>
<tr>
<td>28</td>
<td>0.926</td>
<td>0.653</td>
</tr>
<tr>
<td>29</td>
<td>0.923</td>
<td>0.680</td>
</tr>
<tr>
<td>30</td>
<td>0.921</td>
<td>0.705</td>
</tr>
<tr>
<td>31</td>
<td>0.918</td>
<td>0.729</td>
</tr>
<tr>
<td>32</td>
<td>0.915</td>
<td>0.752</td>
</tr>
<tr>
<td>33</td>
<td>0.913</td>
<td>0.774</td>
</tr>
<tr>
<td>34</td>
<td>0.910</td>
<td>0.794</td>
</tr>
<tr>
<td>35</td>
<td>0.907</td>
<td>0.813</td>
</tr>
<tr>
<td>36</td>
<td>0.904</td>
<td>0.831</td>
</tr>
<tr>
<td>37</td>
<td>0.902</td>
<td>0.848</td>
</tr>
<tr>
<td>38</td>
<td>0.899</td>
<td>0.863</td>
</tr>
<tr>
<td>39</td>
<td>0.896</td>
<td>0.877</td>
</tr>
<tr>
<td>40</td>
<td>0.893</td>
<td>0.891</td>
</tr>
<tr>
<td>41</td>
<td>0.891</td>
<td>0.903</td>
</tr>
<tr>
<td>42</td>
<td>0.888</td>
<td>0.913</td>
</tr>
<tr>
<td>43</td>
<td>0.885</td>
<td>0.923</td>
</tr>
<tr>
<td>44</td>
<td>0.883</td>
<td>0.932</td>
</tr>
<tr>
<td>45</td>
<td>0.880</td>
<td>0.940</td>
</tr>
<tr>
<td>46</td>
<td>0.877</td>
<td>0.948</td>
</tr>
<tr>
<td>47</td>
<td>0.874</td>
<td>0.954</td>
</tr>
<tr>
<td>48</td>
<td>0.872</td>
<td>0.960</td>
</tr>
<tr>
<td>49</td>
<td>0.869</td>
<td>0.965</td>
</tr>
<tr>
<td>50</td>
<td>0.866</td>
<td>0.970</td>
</tr>
</tbody>
</table>
Birthday Paradox

What is the minimum number of people who need to be in a room so that the probability that at least two of them have the same birthday is greater than 50%?

We make the following assumptions: The birthdays of the people in the room are independent, each birthday is equally likely, there are 366 days in the year.

\[ X = \text{“at least two out of } n \text{ people in a room have the same birthday”} \]

\[ P(X) = 1 - P(Y) \]

\[ Y = \text{“all } n \text{ people in the room have different birthdays”} \]

\[ P(Y) = \frac{365}{366} \times \frac{364}{366} \times \frac{363}{366} \times \ldots \times \frac{367-n}{366} \]

For \( n=22 \), \( P(X)=0.475 \).

For \( n=23 \), \( P(X)=0.506 > 50 \% \).

For \( n=47 \), \( P(X)>95 \% \).
Applications

The Pollard-Rho algorithm for the factorization of integers makes use of the Birthday Paradox.

The Birthday Paradox has also applications in the analysis of randomized algorithms, e.g. Hashing.
Why Hashing?

- Hashing-based data structures implement dictionaries (abstract data type that supports insert(x), delete(x), search(x)).
- Hashing is a space-saving *Direct Storage*.

**Direct Storage**

Store key $x$ in table $A$ at position $A[x]$ – with the following disadvantages:
- table can become very large
- even if we should be able to get a table, we will waste a lot of storage when we only store a small number of keys compared to the size of the table

**Hashing**

Use $m$ separate lists instead of one huge list. A „hash function“ transforms every possible key $x$ into a list number $h(x)$ between 0 and $m-1$:

$h(x): U \rightarrow \{0, 1, \ldots, m-1\}$
Collisions

A **collision** occurs when more than one key is assigned to a memory location, i.e. when two keys $x$ and $y \in U, x \neq y$, are hashed on the same value of the hashtable: $h(x) = h(y)$ for $x, y \in U, x \neq y$.

**Chaining** is one of several approaches to augmenting a hash table to resolve collisions. In chaining, each memory location holds a pointer to a linked list, initialized to be empty. When we hash an input to a location, we add the input to the list in that location. Collisions still hurt us, but only in that we take more time traversing a linked list.
How many Collisions?

In the best case, there are no collisions at all, h is injective (static dictionaries can do that with perfect hashing).

In the worst case, all of the n keys are hashed to the exactly same position in the hash table.

Both possibilities are unlikely. We need the following analysis.

1. Average-case analysis
   Choose the keys in a way that the probability for a collision is only 1/m.
   („Randomness is in the user‘s responsibility“)

2. Universal Hashing
   choose the hash function randomly so that the algorithm behaves well for all possible key sets.
   („Randomness is in the algorithm‘s responsibility“)
Average-case analysis

Assumptions:
1. The keys are chosen randomly out of U.
2. \( h \) distributes the keys evenly to the places of the hash table.

What follows:
The number of keys from U that are hashed to a specific value between 0 and \( m-1 \) is

\[
\frac{|U|}{m}
\]

(the keys are hashed in every slot with the same probability).

Therefore:
\[
P(h(x)=h(y)) = \frac{\#\text{hits}}{\#\text{possibilities}} = \frac{|U|/m}{|U|} = 1/m
\]

Costs of an operation: \( O(t+n/m) \) - \( t \) are the costs to compute \( h(x) \)
Universal Hashing (Carter/Fredman 1979)

The hash function \( h: U \rightarrow \{0, \ldots, m-1\} \) is chosen randomly out of the set \( H_0 \) of all possible functions from \( U \) to \( \{0, \ldots, m-1\} \).

What follows:

The number of functions from \( H_0 \) that yield to a collision between two different keys \( x \) and \( y \in U \), is

\[
|H_0|/m
\]

Therefore:

\[
P(h(x)=h(y)) = \text{#hits} / \text{#possibilities} = |H_0|/m / |H_0| = 1/m
\]

Costs of an operation: \( O(t+n/m) \) - \( t \) are the costs to compute \( h(x) \).