Overview

- **Midterm#3 in one week on 4/26**
  - (algo/number/graph/tree), review next lecture.
- **HW7 due on next Tuesday 4/24**
- **Last Lecture:** Completed Graph/Tree
  - Connectivity of Graphs, Shortest Path Problem, Traveling salesman problem, Trees, Spanning Trees, Rooted Trees, Binary Trees
- **This Lecture:** Complete Trees & Counting/Combinatorics
  - Rooted Tree Apps & Tree Representations & Tree Traversal Algorithms
  - Sum and Product rules
  - Principle of Inclusion-Exclusion
  - Tree Diagram
  - Pigeon Hole Principle
  - Permutations
  - Combinations

A child can choose one jellybean out of two jellybeans (red, black), and one gummy bear out of three gummy bears (yellow, green, white). How many different sets of candy can the child have?

Choose jellybean

Choose gummy bear

\[ \therefore 6 \text{ outcomes} \]
• Family tree - not only interesting but also useful for research in medical genetics.
• Files on your computer are organized in a hierarchical (treelike) structure (nested folders).
• Algebraic expression involving binary operations can be represented by a labeled binary tree.

\[(2 + x) - (y * 3)\]

\[2 + (x - y) * 3\]

Binary Tree Representation

Binary trees have special characteristics: the identity of the left and right child.

Binary tree

<table>
<thead>
<tr>
<th>left child</th>
<th>right child</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Left child-right child array representation

Pointer representation
Tree Traversal Algorithms

Traversal of Tree: visit every nodes of a tree in a systematic order

The 3 common tree traversal algorithms are: **preorder**, **inorder**, and **postorder** traversal.

These terms refer to the order in which the root of a tree is visited compared to the subtree nodes.

In these traversal methods, it is helpful to use the recursive view of a tree, where the root of a tree is a parent of the roots of subtrees.
e.g. Do a preorder, inorder, and postorder traversal of the tree.

Preorder: a, b, e, f, c, d, g, i, h
Inorder:  e, b, f, a, c, i, g, d, h
Postorder: e, f, b, c, i, g, h, d, a

Algebraic expressions represented as binary trees

Inorder traversal: (2 + x) * 4
Preorder traversal: * + 2 x 4
Postorder traversal: 2 x + 4 *

- **infix notation**
  operation symbol appears between the 2 operands.

- **prefix notation**
  operation symbol precedes its operands. (Lisp)

- **postfix notation**
  operation symbol follows its operands. (PS)
Chapter 7. Counting: Combinatorics

- **Combinatorics** is a branch of mathematics that deals with counting. Counting is an important basic part of Discrete Math:

  Examples:
  
  “How many possible ways are there to pick 11 soccer players out of a 20-player team?”

  “How many different 8-letter passwords are there?”

  Most importantly, counting is the basis for computing **probabilities** of discrete events.
  
  (“What is the probability of winning the lottery?”)
Basic Counting Principles

• **The sum rule**: If a task \( T_1 \) can be done in \( n_1 \) ways and a second task \( T_2 \) in \( n_2 \) ways, and **if these two tasks cannot be done at the same time**, then there are \( n_1 + n_2 \) ways to do either task.

• **Example**: The department will award a free computer to either a CS student or a CS professor. How many different choices are there, if there are 530 students and 15 professors? There are 530 + 15 = 545 choices.

• **Generalized sum rule**: If we have tasks \( T_1, T_2, \ldots, T_m \) that can be done in \( n_1, n_2, \ldots, n_m \) ways, respectively, and no two of these tasks can be done at the same time, then there are \( n_1 + n_2 + \ldots + n_m \) ways to do one of these tasks.

• **The product rule**: Suppose that a procedure can be broken down into two successive tasks \( T_1 \) and \( T_2 \). If there are \( n_1 \) ways to do the first task and \( n_2 \) ways to do the second task **after the first task has been done**, then there are \( n_1 n_2 \) ways to do the procedure.

• **Example**: How many different license plates are there that containing exactly two English letters? Solution: There are 26 possibilities to pick the first letter, then 26 possibilities for the second one. So there are 26 \cdot 26 = 676 different license plates.

• **Generalized product rule**: If we have a procedure consisting of sequential tasks \( T_1, T_2, \ldots, T_m \) that can be done in \( n_1, n_2, \ldots, n_m \) ways, respectively, then there are \( n_1 \cdot n_2 \cdot \ldots \cdot n_m \) ways to carry out the procedure.
• Example: The last part of your phone # contains 4 digits. How many four-digit numbers are there (ATM PIN#)?
(Select a password of 4 digits, how many possible passwords?)

some 4 digit numbers: 9933, 1982, 2017, 1016, 0007, ...

\[ 10 \times 10 \times 10 \times 10 = 10,000 \text{ different numbers outcomes} \]

• Example: 10 toppings available in a pizza store, and the customer can choose any # of toppings, how many different pizzas the store can make?

<table>
<thead>
<tr>
<th>Toppings</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>choose it</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>...</td>
<td>2</td>
</tr>
<tr>
<td>or w/o it</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

or w/o it: \[ 2 \times 2 \times 2 \times ... \times 2 = 2^{10} = 1,024 \]

• The sum and product rules can also be phrased in terms of set theory.

**Sum rule:** Let \( A_1, A_2, ..., A_m \) be disjoint sets. Then the number of ways to choose any element from one of these sets is
\[
| A_1 \cup A_2 \cup ... \cup A_m | = | A_1 | + | A_2 | + ... + | A_m |
\]

**Product rule:** Let \( A_1, A_2, ..., A_m \) be finite sets. Then the number of ways to choose one element from each set in the order \( A_1, A_2, ..., A_m \) is
\[
| A_1 \times A_2 \times \ldots \times A_m | = | A_1 | \cdot | A_2 | \cdot \ldots \cdot | A_m |
\]

• Next, we look at various methods to compute results using combination of sum rule and product rule.
• Example: How many ways to choose two courses from CSC212, CSC210, CSC211. If 6 sections of CSC212, 7 sections of CSC210 and 5 sections of CSC211 are offered.
  \[
  \begin{align*}
  \text{# of ways to select CSC212 & CSC210} & : 6 \times 7 = 42 \\
  \text{# of ways to select CSC212 & CSC211} & : 6 \times 5 = 30 \\
  \text{# of ways to select CSC210 & CSC211} & : 7 \times 5 = 35 \\
  \text{total: 107 ways}
  \end{align*}
  \]

• Example: How many 3-letter words have a letter repeated if words are formed from \{a,b,c,d,e\}.
  a. Total # of 3-letter words : 5 \times 5 \times 5 = 125
  b. Total # of 3-letter words with distinct letters : 5 \times 4 \times 3 = 60
  c. Total # of words with repeated letters: (a)-(b) = 65
  Note: counting the complement

Tree Diagrams

• How many bit strings of length four do not have two consecutive 1s?

<table>
<thead>
<tr>
<th>1st bit</th>
<th>2nd bit</th>
<th>3rd bit</th>
<th>4th bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

There are 8 strings.
Recall Principle of Inclusion-Exclusion

- Recall this in set theory: \(|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|\)

- How many bit strings of length 8 either start with a 1 or end with 00?

  **Task 1:** Construct a string of length 8 that starts with a 1.
  \(1*2*2*2*2*2*2*2\)
  **Product rule:** Task 1 can be done in \(1 \cdot 2^7 = 128\) ways.

  **Task 2:** Construct a string of length 8 that ends with 00.
  \(2*2*2*2*2*2*1*1\)
  **Product rule:** Task 2 can be done in \(2^6 = 64\) ways.

  **Task 3:** How many strings start with 1 and end with 00?

  There is one way to pick the first bit (1),
  two ways for the second, ..., sixth bit (0 or 1),
  one way for the seventh, eighth bit (0).
  \(1*2*2*2*2*2*1*1\)
  **Product rule:** In \(2^5 = 32\) cases

  Total number of strings is \(128+64-32 = 160\)
More examples:
How many bit strings of length 8? \(2^8 = 256\)
How many bit strings of length 8 with no 1? 1
How many bit strings of length 8 with exactly one 1? 8
How many bit strings of length 8 with exactly two 1s?
1_ _ _ _ _ _ _ // 1st position and 7 other positions
_ 1 _ _ _ _ _ _ // 2nd position and 6 other positions
...
7+6+5+4+3+2+1=28 or \(C(8,2)\) // will cover \(C()\) later
How many bit strings of length 8 with at most two 1s?
1+8+28=37
How many bit strings of length 8 with at least three 1s?
256-37=219

Recall The Pigeonhole Principle
Example: A bank requires all customer to choose a four-digit code to use with an ATM card. The code must contain two letters in first two positions and two numbers in last two positions. The bank has 75000 customers. Show that at least two customers choose the same four-digit code.

Total number of codes: \(26*26*10*10 = 67600\).

By the Pigeonhole Principle, there are at least two customers choose the same code.
**Permutations**

- A **permutation** is an ordered arrangement of objects. (without repetition)

- An ordered arrangement of \( r \) elements \( (r < |S|) \) of a set \( S \) is called an **\( r \)-permutation**

- **Example**: Let \( S = \{1, 2, 3\} \).
  
  \((3,1,2)\) or \((2,1,3)\) are two different examples of permutation. \((3,2)\) or \((1,3)\) or \((2,3)\) are three examples of 2-permutation.

---

\[ \begin{align*}
\text{The number of } r \text{-permutations of a set with } n \text{ distinct elements is denoted by } P(n, r). \\
\text{We can calculate } P(n, r) \text{ with the product rule: } P(n, r) &= n \cdot (n-1) \cdot (n-2) \cdots (n-r+1). \\
\text{(} n \text{ choices for the first element, } (n-1) \text{ for the second one, } (n-2) \text{ for the third one...)}
\end{align*} \]

**General formula**: \( P(n, r) = \frac{n!}{(n-r)!} \)

Note:  
\( P(n, 0) = \frac{n!}{(n-0)!} = 1 \)  
\( P(n, 1) = \frac{n!}{(n-1)!} = n \)  
\( P(n, n) = \frac{n!}{(n-n)!} = n! \; \because 0! = 1 \)
• Example:

\[ P(8, 3) = 8 \cdot 7 \cdot 6 = 336 = (8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1)/(5 \cdot 4 \cdot 3 \cdot 2 \cdot 1) \]

• Example: How many 3-letter words (not necessarily meaningful) can be formed from the word “Compiler” if no letter can be repeated?

\[ P(8, 3) = \frac{8!}{5!} = 8 \cdot 7 \cdot 6 = 336 \]

• Example: How many ways can a president and vice-president be selected from a group of 20 people?

\[ P(20, 2) = \frac{20!}{18!} = 20 \cdot 19 = 380 \]

Combinations

• Example: How many ways are there to pick a set of 3 people from a group of 6?

There are 6 choices for the first person, 5 for the second one, and 4 for the third one, so there are 6 \cdot 5 \cdot 4 = 120 ways to do this.

This is not the correct result!

For example, picking person C, then person A, and then person E leads to the same group as first picking E, then C, and then A.

However, these cases are counted separately in the above formulation.

So how can we compute how many different subsets of people can be picked (i.e., we want to disregard the order of picking)?
• An \( r \)-combination of elements of a set is an unordered selection of \( r \) elements from the set. Thus, an \( r \)-combination is simply a subset of the set with \( r \) elements.

• Example: Let \( S = \{1, 2, 3, 4\} \). Then \( \{1, 3, 4\} = \{3, 1, 4\} \) is an example member of 3-combination.

• The number of all possible \( r \)-combinations from a set with \( n \) distinct elements is denoted by \( C(n, r) \) (“\( n \) choose \( r \)”).

• Example: \( C(4, 2) = 6 \), i.e. the 2-combinations of a set \( \{1, 2, 3, 4\} \) are \( \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\} \).

• How can we derive \( C(n, r) \)? Formula?

Consider that we can obtain \( P(n, r) \) of a set in the following way:
First, find all \( r \)-combinations of the set, i.e. \( C(n, r) \)
Then, for each \( r \)-combination, generate all possible orderings, i.e. \( P(r, r) = r! \). Therefore, we have: \( P(n, r) = C(n, r)^*P(r, r) \)

\[
C(n, r) = \frac{P(n, r)}{P(r, r)} = \frac{n!}{(n-r)!} \div \frac{r!}{(r-r)!} = \frac{n!}{r!(n-r)!}
\]

• Now we can answer our question: How many ways are there to pick a set of 3 people from a group of 6 (disregarding the order of picking)?

\( C(6, 3) = 6!/3!3! = 720/6 = 20 \)

• Corollary: Let \( n \) and \( r \) be nonnegative integers with \( r \leq n \). Then \( C(n, r) = C(n, n-r) \).

Note: each choice \( r \)-elements determine a unique choice of \((n-r)\)-elements
• Example: A soccer club has 8 female and 7 male members. For today’s match, the coach wants to have 6 female and 5 male players on the grass. How many possible configurations are there? 
   Ans: $C(8, 6) \times C(7, 5) = 8!/6! \times 7!/5! = 28 \times 21 = 588$

• Example: A committee of 8 students is to be selected from a class consisting of 19 freshmen and 34 sophomores.
   In how many ways can 3 freshmen and 5 sophomores be selected? Ans: $C(19,3) \times C(34,5)$
   In how many ways can a committee with exactly 1 freshman be selected? Ans: $C(19,1) \times C(34,7)$
   In how many ways can a committee with at most 1 freshman be selected? 0 freshman + 1 freshman = $C(34,8) + C(19,1) \times C(34,7)$
   In how many ways can a committee with at least 1 freshman be selected? All 8-combination – no freshman = $C(53,8) - C(34,8)$

Summary:

$C(n, r) = \frac{n!}{(n-r)!r!} = \frac{P(n, r)}{r!} = C(n, r)$

Note:
- $C(n, 0) = \frac{n!}{(n-0)!} = 1$
- $C(n, 1) = \frac{n!}{(n-1)!!} = n$
- $C(n, n) = \frac{n!}{(n-n)!n!} = 1$

$C(n, r) = C(n, n-r)$.