Foundations of Computer Science
Lecture 13

Counting

Counting Sequences
Build-Up Counting
Counting One Set by Counting Another: Bijection
Permutations and Combinations
Be careful of what you read in the media: sex in America.

Bipartite Graphs and Matching (Hall’s Theorem).

Stable Marriage.

Conflict Graphs and Coloring.
  ▶ Every tree is 2-colorable.

Other Graph Problems
  ▶ Connected components, spanning tree, Euler paths, network flow. ([EASY])
  ▶ Hamiltonian paths, \( K \)-center, vertex cover, dominating set. ([HARD])
Today: Counting

2. Build-up counting.
3. Counting one set by counting another: bijection.
4. Permutations and combinations.

**Fun Counting Fact.** Radio took 38 years to reach 50M users. TV took 13 years. The World Wide Web took just 4 years.
Three colors of candy:

\[
\text{red}, \text{blue}, \text{green}.
\]

A goody-bag has 3 candies. How many distinct goody-bags?

(Only the number of each color matters: \{\text{red, red, blue}\} and \{\text{red, blue, red}\} are the “same” goody-bag.)
Three colors of candy:

\[
\text{\color{red}●}, \text{\color{blue}●}, \text{\color{green}●}.
\]

A goody-bag has 3 candies. How many distinct goody-bags?
(Only the number of each color matters: \{\color{red}●●●\} and \{\color{blue}●●●\} are the “same” goody-bag.)
Discrete Math Is About Objects We Can Count

Three colors of candy:

\[ \bullet, \bullet, \bullet. \]

A goody-bag has 3 candies. How many distinct goody-bags?
(Only the number of each color matters: \{\bullet\bullet\bullet\} and \{\bullet\bullet\red\bullet\} are the “same” goody-bag.)

\[
\{\bullet\bullet\bullet\} \quad \{\bullet\bullet\red\bullet\} \quad \{\bullet\bullet\Green\bullet\} \quad \{\bullet\blue\bullet\bullet\} \quad \{\bullet\bullet\Green\bullet\blue\bullet\} \quad \{\bullet\red\bullet\bullet\bullet\} \quad \{\bullet\bullet\bullet\red\bullet\\} \quad \{\bullet\bullet\bullet\bullet\blue\bullet\} \quad \{\bullet\bullet\bullet\bullet\bullet\bullet\} \]

**Challenge Problems.**

1. What if there are 5 candies per goody-bag and 10 colors of candy?
2. Goody-bags come in bulk packs of 5. How many different bulk packs are there?

There are too many to list out. We need tools!
How many binary sequences of length 3: \{000, 001, 010, 011, 100, 101, 110, 111\}. 
Sum Rule

How many binary sequences of length 3: \{000, 001, 010, 011, 100, 101, 110, 111\}.

There are two types: those ending in 0 and those ending in 1,

\[ \{b_1b_2b_3\} = \{b_1b_2 \cdot 0\} \cup \{b_1b_2 \cdot 1\} \]
How many binary sequences of length 3: \{000, 001, 010, 011, 100, 101, 110, 111\}.

There are two types: those ending in 0 and those ending in 1,
\[\{b_1b_2b_3\} = \{b_1b_2 \cdot 0\} \cup \{b_1b_2 \cdot 1\}\]

**Sum Rule.** $N$ objects of two types: $N_1$ of type-1 and $N_2$ of type-2. Then,
\[N = N_1 + N_2.\]
How many binary sequences of length 3: \{000, 001, 010, 011, 100, 101, 110, 111\}.

There are two types: those ending in 0 and those ending in 1,

\[ \{b_1b_2b_3\} = \{b_1b_2 \cdot 0\} \cup \{b_1b_2 \cdot 1\} \]

**Sum Rule.** \(N\) objects of two types: \(N_1\) of type-1 and \(N_2\) of type-2. Then,

\[ N = N_1 + N_2. \]

\[ |\{b_1b_2b_3\}| = |\{b_1b_2 \cdot 0\}| + |\{b_1b_2 \cdot 1\}| \quad \text{(sum rule)} \]
How many binary sequences of length 3: \{000, 001, 010, 011, 100, 101, 110, 111\}.

There are two types: those ending in 0 and those ending in 1,
\[ \{b_1b_2b_3\} = \{b_1b_2 \cdot 0\} \cup \{b_1b_2 \cdot 1\} \]

**Sum Rule.** *N* objects of two types: *N*₁ of type-1 and *N*₂ of type-2. Then,
\[ N = N_1 + N_2. \]

\[
|\{b_1b_2b_3\}| = |\{b_1b_2 \cdot 0\}| + |\{b_1b_2 \cdot 1\}| \quad \text{(sum rule)}
\]
\[ = |\{b_1b_2\}| \times 2 \]
Sum Rule

How many binary sequences of length 3: \{000, 001, 010, 011, 100, 101, 110, 111\}.

There are two types: those ending in 0 and those ending in 1,

\[ \{b_1b_2b_3\} = \{b_1b_2 \cdot 0\} \cup \{b_1b_2 \cdot 1\} \]

**Sum Rule.** \(N\) objects of two types: \(N_1\) of type-1 and \(N_2\) of type-2. Then,

\[ N = N_1 + N_2. \]

\[
|\{b_1b_2b_3\}| = |\{b_1b_2 \cdot 0\}| + |\{b_1b_2 \cdot 1\}| \\
= |\{b_1b_2\}| \times 2 \\
= (|\{b_1 \cdot 0\}| + |\{b_1 \cdot 1\}|) \times 2
\]

(sum rule)
How many binary sequences of length 3: \{000, 001, 010, 011, 100, 101, 110, 111\}.

There are two types: those ending in 0 and those ending in 1,
\[\{b_1b_2b_3\} = \{b_1b_2 \cdot 0\} \cup \{b_1b_2 \cdot 1\}\]

**Sum Rule.** \(N\) objects of two types: \(N_1\) of type-1 and \(N_2\) of type-2. Then,
\[N = N_1 + N_2.\]

\[
|\{b_1b_2b_3\}| = |\{b_1b_2 \cdot 0\}| + |\{b_1b_2 \cdot 1\}| \quad \text{(sum rule)}
\]
\[= |\{b_1b_2\}| \times 2 \]
\[= (|\{b_1 \cdot 0\}| + |\{b_1 \cdot 1\}|) \times 2 \quad \text{(sum rule)}
\]
\[= |\{b_1\}| \times 2 \times 2\]
How many binary sequences of length 3: \{000, 001, 010, 011, 100, 101, 110, 111\}.

There are two types: those ending in 0 and those ending in 1,
\[
\{b_1b_2b_3\} = \{b_1b_2 \cdot 0\} \cup \{b_1b_2 \cdot 1\}
\]

**Sum Rule.** \(N\) objects of two types: \(N_1\) of type-1 and \(N_2\) of type-2. Then,
\[
N = N_1 + N_2.
\]

\[
\begin{align*}
|\{b_1b_2b_3\}| &= |\{b_1b_2 \cdot 0\}| + |\{b_1b_2 \cdot 1\}| & \text{(sum rule)} \\
&= |\{b_1b_2\}| \times 2 \\
&= (|\{b_1 \cdot 0\}| + |\{b_1 \cdot 1\}|) \times 2 & \text{(sum rule)} \\
&= |\{b_1\}| \times 2 \times 2 \\
&= 2 \times 2 \times 2
\end{align*}
\]
Product Rule

\[ |\{ b_1 \ b_2 \ b_3 \}| \]
\[ 2 \times 2 \times 2 \]

Let \( N \) be the number of choices for a sequence
\[ x_1x_2x_3 \cdots x_{r-1}x_r. \]
Let $N$ be the number of choices for a sequence $x_1x_2x_3 \cdots x_{r-1}x_r$. 

Let $N_1$ be the number of choices for $x_1$;
\[ |\{b_1 \; b_2 \; b_3\}| \]
\[ 2 \times 2 \times 2 \]

Let \( N \) be the number of choices for a sequence
\[ x_1x_2x_3 \cdots x_{r-1}x_r. \]

Let \( N_1 \) be the number of choices for \( x_1 \);
Let \( N_2 \) be the number of choices for \( x_2 \) after you choose \( x_1 \);
Let $N$ be the number of choices for a sequence

$$x_1x_2x_3 \cdots x_{r-1}x_r.$$ 

Let $N_1$ be the number of choices for $x_1$;
Let $N_2$ be the number of choices for $x_2$ after you choose $x_1$;
Let $N_3$ be the number of choices for $x_3$ after you choose $x_1x_2$;
Let $N$ be the number of choices for a sequence 
\[ x_1 x_2 x_3 \cdots x_{r-1} x_r. \]

Let $N_1$ be the number of choices for $x_1$;
Let $N_2$ be the number of choices for $x_2$ after you choose $x_1$;
Let $N_3$ be the number of choices for $x_3$ after you choose $x_1 x_2$;
Let $N_4$ be the number of choices for $x_4$ after you choose $x_1 x_2 x_3$;
Let $N$ be the number of choices for a sequence $x_1x_2x_3\cdots x_{r-1}x_r$.

Let $N_1$ be the number of choices for $x_1$;
Let $N_2$ be the number of choices for $x_2$ after you choose $x_1$;
Let $N_3$ be the number of choices for $x_3$ after you choose $x_1x_2$;
Let $N_4$ be the number of choices for $x_4$ after you choose $x_1x_2x_3$;
\vdots
Let $N_r$ be the number of choices for $x_r$ after you choose $x_1x_2x_3\cdots x_{r-1}$.

\[
\left| \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} \right| = 2 \times 2 \times 2
\]
Product Rule

\[ |\{b_1 \ b_2 \ b_3\}| \]

\[ 2 \times 2 \times 2 \]

Let \( N \) be the number of choices for a sequence

\[ x_1 x_2 x_3 \cdots x_{r-1} x_r. \]

Let \( N_1 \) be the number of choices for \( x_1 \);
Let \( N_2 \) be the number of choices for \( x_2 \) \textit{after you choose} \( x_1 \);
Let \( N_3 \) be the number of choices for \( x_3 \) \textit{after you choose} \( x_1 x_2 \);
Let \( N_4 \) be the number of choices for \( x_4 \) \textit{after you choose} \( x_1 x_2 x_3 \);
\[ \vdots \]
Let \( N_r \) be the number of choices for \( x_r \) \textit{after you choose} \( x_1 x_2 x_3 \cdots x_{r-1} \).

\[ N = N_1 \times N_2 \times N_3 \times N_4 \times \cdots \times N_r. \]
Let $N$ be the number of choices for a sequence $x_1 x_2 x_3 \cdots x_{r-1} x_r$.

Let $N_1$ be the number of choices for $x_1$;
Let $N_2$ be the number of choices for $x_2$ after you choose $x_1$;
Let $N_3$ be the number of choices for $x_3$ after you choose $x_1 x_2$;
Let $N_4$ be the number of choices for $x_4$ after you choose $x_1 x_2 x_3$;
::
Let $N_r$ be the number of choices for $x_r$ after you choose $x_1 x_2 x_3 \cdots x_{r-1}$.

$$N = N_1 \times N_2 \times N_3 \times N_4 \times \cdots \times N_r.$$ 

**Example.** There are $2^n$ binary sequences of length $n$: $N_1 = N_2 = \cdots = N_n = 2$.

The sum and product rules are the only basic tools we need ... plus TINKERING.
**Examples**

### Menus.

- **breakfast**: \( \in \{\text{pancake, waffle, Doritos}\} \)
- **lunch**: \( \in \{\text{burger, Doritos}\} \)
- **dinner**: \( \in \{\text{salad, steak, Doritos}\} \)
Menus. breakfast ∈ \{pancake, waffle, Doritos\}  
lunch ∈ \{burger, Doritos\}  
dinner ∈ \{salad, steak, Doritos\}  

\[|\{BLD\}| = 3 \times 2 \times 3 = 18.\]
Examples

Menus. \( \text{breakfast} \in \{\text{pancake, waffle, Doritos}\} \) \( \text{midday} \in \{\text{burger, Doritos}\} \) \( \text{dinner} \in \{\text{salad, steak, Doritos}\} \)

(\( \bullet \) every menu is a sequence \( BLD \) and \( \bullet \) every sequence \( BLD \) is a unique menu.)

\(|\{BLD\}| = 3 \times 2 \times 3 = 18.\)
Examples

1. **Menus.**
   - breakfast $\in \{\text{pancake, waffle, Doritos}\}$
   - lunch $\in \{\text{burger, Doritos}\}$
   - dinner $\in \{\text{salad, steak, Doritos}\}$

   ($\bullet$ every menu is a sequence $BLD$ and $\bullet$ every sequence $BLD$ is a unique menu.)

   $|\{BLD\}| = 3 \times 2 \times 3 = 18.$

2. **NY plates.** $|\{ABC-1234\}|$
Examples

1. **Menus.**
   - breakfast $\in \{\text{pancake, waffle, Doritos}\}$
   - lunch $\in \{\text{burger, Doritos}\}$
   - dinner $\in \{\text{salad, steak, Doritos}\}$

   ($\bullet$ every menu is a sequence $BLD$ and $\bullet$ every sequence $BLD$ is a unique menu.)

   $|\{BLD\}| = 3 \times 2 \times 3 = 18.$

2. **NY plates.**
   - $|\{\text{ABC-1234}\}| = 26 \times 26 \times 26 \times 10 \times 10 \times 10 \times 10 \approx 176$ million.
**Examples**

1. **Menus.**  
   - Breakfast $\in \{\text{pancake, waffle, Doritos}\}$  
   - Lunch $\in \{\text{burger, Doritos}\}$  
   - Dinner $\in \{\text{salad, steak, Doritos}\}$  
   
   (every menu is a sequence $BLD$ and every sequence $BLD$ is a unique menu.)  
   
   $|\{BLD\}| = 3 \times 2 \times 3 = 18.$

2. **NY plates.**  
   
   $|\{\text{ABC-1234}\}| = 26 \times 26 \times 26 \times 10 \times 10 \times 10 \times 10 \approx 176$ million.

3. **Races.** With 10 runners, how many top-3 finishes?
Examples

1. **Menus.**
   - breakfast $\in \{$pancake, waffle, Doritos$\}$
   - lunch $\in \{$burger, Doritos$\}$
   - dinner $\in \{$salad, steak, Doritos$\}$

   $(\bullet$ every menu is a sequence $BLD$ and $\bullet$ every sequence $BLD$ is a unique menu$.)$

   $|\{BLD\}| = 3 \times 2 \times 3 = 18$.

2. **NY plates.**

   $|\{ABC-1234\}| = 26 \times 26 \times 26 \times 10 \times 10 \times 10 \times 10 \approx 176$ million.

3. **Races.**

   With $10$ runners, how many top-3 finishes?

   $|\{FST\}| = 10 \times 9 \times 8 = 720$. 
Examples

Menus.  breakfast $\in \{\text{pancake, waffle, Doritos}\}$  $|\{BLD\}| = 3 \times 2 \times 3 = 18.$
  lunch $\in \{\text{burger, Doritos}\}$
  dinner $\in \{\text{salad, steak, Doritos}\}$
  ($\bullet$ every menu is a sequence $BLD$ and $\bullet$ every sequence $BLD$ is a unique menu.)

NY plates.  $|\{\text{ABC-1234}\}| = 26 \times 26 \times 26 \times 10 \times 10 \times 10 \times 10 \approx 176$ million.

Races. With 10 runners, how many top-3 finishes?  $|\{FST\}| = 10 \times 9 \times 8 = 720.$

Passwords. Use: $\{a,\ldots,z\}, \{A,\ldots,Z\}, \{0,\ldots,9\}$, special: $\{!,@,#,$,$\%,\&,*,(,)\}$.  Rules: Length is 8. Must have at least one special.
  $|\{\text{passwords}\}| = 72 \times 72 \times \cdots \times 72 = 72^8$ (product rule)
Examples

1. **Menus.** breakfast ∈ \{pancake, waffle, Doritos\}  \quad \text{\{BLD\}} = 3 \times 2 \times 3 = 18.
   
   lunch ∈ \{burger, Doritos\}
   
   dinner ∈ \{salad, steak, Doritos\}
   
   (• every menu is a sequence BLD and • every sequence BLD is a unique menu.)

2. **NY plates.** |\{ABC-1234\}| = 26 \times 26 \times 26 \times 10 \times 10 \times 10 \times 10 \approx 176 \text{ million}.

3. **Races.** With 10 runners, how many top-3 finishes? |\{FST\}| = 10 \times 9 \times 8 = 720.

4. **Passwords.** Use: \{a, \ldots, z\}, \{A, \ldots, Z\}, \{0, \ldots, 9\}, special: \{!,@,#,$,%,*),(,),}.
   
   Rules: Length is 8. Must have at least one special.
   
   |\{passwords\}| = 72 \times 72 \times \cdots \times 72 = 72^8 \quad \text{(product rule)}
   
   = |\{valid\}| + |\{invalid\}| \quad \text{(sum rule)}
**Examples**

1. **Menus.**
   - breakfast $\in \{\text{pancake, waffle, Doritos}\}$
   - lunch $\in \{\text{burger, Doritos}\}$
   - dinner $\in \{\text{salad, steak, Doritos}\}$
   - \( |\{\text{BLD}\}| = 3 \times 2 \times 3 = 18 \).
   - (• every menu is a sequence BLD and • every sequence BLD is a unique menu.)

2. **NY plates.**
   - \( |\{\text{ABC-1234}\}| = 26 \times 26 \times 26 \times 10 \times 10 \times 10 \approx 176 \text{ million} \).

3. **Races.** With 10 runners, how many top-3 finishes?
   - \( |\{\text{FST}\}| = 10 \times 9 \times 8 = 720 \).

4. **Passwords.**
   - Use: \{a,...,z\}, \{A,...,Z\}, \{0,...,9\}, special: \{!,@,#,$,%,$,\&,*,(,\}.
   - Rules: Length is 8. Must have at least one special.
   - \( |\{\text{passwords}\}| = 72 \times 72 \times \cdots \times 72 = 72^8 \)  
     \( \text{(product rule)} \)
   - \( = |\{\text{valid}\}| + |\{\text{invalid}\}| \) \( \text{(sum rule)} \)
   - \( = |\{\text{valid}\}| + 62^8 \) \( \text{(product rule)} \)
Examples

Menus. \( \text{breakfast} \in \{\text{pancake, waffle, Doritos}\} \quad |\{BLD\}| = 3 \times 2 \times 3 = 18. \)
\( \text{lunch} \in \{\text{burger, Doritos}\} \)
\( \text{dinner} \in \{\text{salad, steak, Doritos}\} \)
\( \text{(● every menu is a sequence } BLD \text{ and ● every sequence } BLD \text{ is a unique menu.)} \)

NY plates. \( |\{ABC-1234\}| = 26 \times 26 \times 26 \times 10 \times 10 \times 10 \approx 176 \text{ million.} \)

Races. With 10 runners, how many top-3 finishes? \( |\{FST\}| = 10 \times 9 \times 8 = 720. \)

Passwords. Use: \( \{a, \ldots, z\}, \{A, \ldots, Z\}, \{0, \ldots, 9\} \), special: \( \{!, @, #, $, %, ^, &, *, (, )\} \).
\( \text{Rules: Length is 8. Must have at least one special.} \)
\[ |\{\text{passwords}\}| = 72 \times 72 \times \cdots \times 72 = 72^8 \quad \text{(product rule)} \]
\[ = |\{\text{valid}\}| + |\{\text{invalid}\}| \quad \text{(sum rule)} \]
\[ = |\{\text{valid}\}| + 62^8 \quad \text{(product rule)} \]
\[ |\{\text{valid}\}| = 72^8 - 62^8 \approx 5 \times 10^{14}. \quad \text{(1 millisecond to test \( \rightarrow \) about 6 months on 32K cores.)} \]
Examples

1. **Menus.**
   
   breakfast ∈ \{pancake, waffle, Doritos\}
   lunch ∈ \{burger, Doritos\}
   dinner ∈ \{salad, steak, Doritos\}
   
   (● every menu is a sequence \textit{BLD} and ● every sequence \textit{BLD} is a \textit{unique} menu.)

   \[|\{\textit{BLD}\}| = 3 \times 2 \times 3 = 18.\]

2. **NY plates.**
   \[|\text{ABC-1234}| = 26 \times 26 \times 26 \times 10 \times 10 \times 10 \times 10 \approx 176 \text{ million}.\]

3. **Races.**
   
   With 10 runners, how many top-3 finishes? 
   \[|\{\textit{FST}\}| = 10 \times 9 \times 8 = 720.\]

4. **Passwords.**
   
   Use: \{a, \ldots , z\}, \{A, \ldots , Z\}, \{0, \ldots , 9\}, special: \{!, @, #, $, %, \&, *, (, )\}.
   
   Rules: Length is 8. Must have at least one special.
   
   \[|\{\text{passwords}\}| = 72 \times 72 \times \cdots \times 72 = 72^8 \quad \text{(product rule)}\]
   \[= |\{\text{valid}\}| + |\{\text{invalid}\}| \quad \text{(sum rule)}\]
   \[= |\{\text{valid}\}| + 62^8 \quad \text{(product rule)}\]
   \[|\{\text{valid}\}| = 72^8 - 62^8 \approx 5 \times 10^{14}. \text{ (1 millisecond to test \rightarrow about 6 months on 32K cores.)}\]

5. **Committees.**
   
   10 students. How many ways to form a party planning committee?
**Examples**

1. **Menus.**
   \[
   \begin{align*}
   \text{breakfast} & \in \{\text{pancake, waffle, Doritos}\} \\
   \text{lunch} & \in \{\text{burger, Doritos}\} \\
   \text{dinner} & \in \{\text{salad, steak, Doritos}\}
   \end{align*}
   \]
   \[
   |\{\text{BLD}\}| = 3 \times 2 \times 3 = 18.
   \]
   (• every menu is a sequence \text{BLD} and • every sequence \text{BLD} is a unique menu.)

2. **NY plates.**
   \[
   |\{\text{ABC-1234}\}| = 26 \times 26 \times 26 \times 10 \times 10 \times 10 \times 10 \approx 176 \text{ million}.
   \]

3. **Races.**
   With 10 runners, how many top-3 finishes?
   \[
   |\{\text{FST}\}| = 10 \times 9 \times 8 = 720.
   \]

4. **Passwords.**
   Use: \{a,...,z\}, \{A,...,Z\}, \{0,...,9\}, special: \{!,@,#,$,%,∧,&,*,(,\}.
   Rules: Length is 8. Must have at least one special.
   \[
   |\{\text{passwords}\}| = 72 \times 72 \times \cdots \times 72 = 72^8 \quad \text{(product rule)}
   \]
   \[
   = |\{\text{valid}\}| + |\{\text{invalid}\}| \quad \text{(sum rule)}
   \]
   \[
   = |\{\text{valid}\}| + 62^8 \quad \text{(product rule)}
   \]
   \[
   |\{\text{valid}\}| = 72^8 - 62^8 \approx 5 \times 10^{14}. \text{ (1 millisecond to test → about 6 months on 32K cores.)}
   \]

5. **Committees.**
   10 students. How many ways to form a party planning committee?
   Each student can be in or out of the committee:
   \[
   2 \times 2 \times \cdots \times 2 = 2^{10} = 1024.
   \]
Examples

1. **Menus.**
   
   breakfast ∈ \{pancake, waffle, Doritos\} \\
   lunch ∈ \{burger, Doritos\} \\
   dinner ∈ \{salad, steak, Doritos\} \\
   (● every menu is a sequence BLD and ● every sequence BLD is a unique menu.) \\
   \[|\{BLD\}| = 3 \times 2 \times 3 = 18.\]

2. **NY plates.**
   
   \[|\{ABC-1234\}| = 26 \times 26 \times 26 \times 10 \times 10 \times 10 \approx 176 \text{ million}.\]

3. **Races.**
   
   With 10 runners, how many top-3 finishes? 
   
   \[|\{FST\}| = 10 \times 9 \times 8 = 720.\]

4. **Passwords.**
   
   Use: \{a, \ldots, z\}, \{A, \ldots, Z\}, \{0, \ldots, 9\}, special: \{!, @, #, $, %, &, *, (, )\}.
   
   Rules: Length is 8. Must have at least one special.
   
   \[|\{\text{passwords}\}| = 72 \times 72 \times \cdots \times 72 = 72^8 \quad \text{(product rule)}\]
   
   \[= |\{\text{valid}\}| + |\{\text{invalid}\}| \quad \text{(sum rule)}\]
   
   \[= |\{\text{valid}\}| + 62^8 \quad \text{(product rule)}\]
   
   \[|\{\text{valid}\}| = 72^8 - 62^8 \approx 5 \times 10^{14}. \text{ (1 millisecond to test → about 6 months on 32K cores.)}\]

5. **Committees.**
   
   10 students. How many ways to form a party planning committee?
   Each student can be in or out of the committee: 
   \[2 \times 2 \times \cdots \times 2 = 2^{10} = 1024.\]
   
   \[|\{\text{committees}\}| = |\{10\text{-bit binary strings}\}| \quad \text{e.g.} \quad \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\} \leftrightarrow \{s_1, s_2, s_4, s_9\} \leftrightarrow \varnothing\]
Challenge Exercises:

- A problem from the area of natural language processing (NLP).

Wikipedia has 40 million articles (6 million in English). You compute an “edit distance” between every pair of articles and store this data in a $40\text{million} \times 40\text{million}$ array of 64-bit double precision numbers.

- Estimate how much RAM you will need to load this matrix into memory. (answer: $\sim 13\text{TB}$)
- Any suggestions on feasibly performing tasks using this edit-distance data?
Challenge Exercises:

• A problem from the area of natural language processing (NLP).
  
  Wikipedia has 40 million articles (6 million in English). You compute an “edit
distance” between every pair of articles and store this data in a 40million × 40million
array of 64-bit double precision numbers.

  ◄ Estimate how much RAM you will need to load this matrix into memory. (answer: \(\sim 13\)TB)
  ◄ Any suggestions on feasibly performing tasks using this edit-distance data?

• A problem in scheduling exams.

  RPI has about 400 courses and 5000 students who induce conflicts among these
courses for exam scheduling. Are 15 exam slots enough to schedule all the exams?
You realize that you need to color a 400-node conflict graph using 15 colors.

  ◄ You can generate and test validity of a 15-coloring at a rate of 3 million per second. Estimate how
long it would take to determine that 15 exam slots is not enough. (answer: \(\sim 10^{456}\)years)
  ◄ Any suggestions on what to do?
Build-Up Counting

Number of binary sequences of length $n$: $2^n$. 

Goody Bags →
Build-Up Counting

Number of binary sequences of length $n$: $2^n$.

$$\binom{n}{k} = \text{number binary sequences of length } n \text{ with exactly } k \text{ 1's} \quad 0 \leq k \leq n.$$
Build-Up Counting

Number of binary sequences of length $n$: $2^n$.

$$\binom{n}{k} = \text{number binary sequences of length } n \text{ with exactly } k \text{ 1's } \quad 0 \leq k \leq n.$$  

Length-3 sequences:

000 001 010 011 100 101 110 111
Build-Up Counting

Number of binary sequences of length $n$: $2^n$.

$\binom{n}{k} = \text{number binary sequences of length } n \text{ with exactly } k \text{ 1's } \quad 0 \leq k \leq n.$

**Length-3 sequences:**

000 001 010 011 100 101 110 111

<table>
<thead>
<tr>
<th>$(n)$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Build-Up Counting

Number of binary sequences of length $n$: $2^n$.

$$\binom{n}{k} = \text{number binary sequences of length } n \text{ with exactly } k \text{ 1's} \quad 0 \leq k \leq n.$$  

<table>
<thead>
<tr>
<th>Length-3 sequences:</th>
<th>$\binom{n}{k}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>000 001 010 011 100 101 110 111</td>
<td>0 1</td>
</tr>
<tr>
<td>Length-4 sequences:</td>
<td>$\binom{n}{k}$</td>
</tr>
<tr>
<td>0000 0001 0010 0011 0100 0101 0110 0111</td>
<td>1 1 1</td>
</tr>
<tr>
<td>1000 1001 1010 1011 1100 1101 1110 1111</td>
<td>$2 \ 1 \ 2 \ 1$</td>
</tr>
<tr>
<td>Length-5 sequences:</td>
<td>$\binom{n}{k}$</td>
</tr>
<tr>
<td>00000 00001 00010 00011 00100 00101 00110 00111</td>
<td>3 1 3 3 1</td>
</tr>
<tr>
<td>01000 01001 01010 01011 01100 01101 01110 01111</td>
<td>4</td>
</tr>
<tr>
<td>10000 10001 10010 10011 10100 10101 10110 10111</td>
<td>5</td>
</tr>
<tr>
<td>11000 11001 11010 11011 11100 11101 11110 11111</td>
<td>6</td>
</tr>
<tr>
<td>11100 11101 11110 11111</td>
<td>7</td>
</tr>
<tr>
<td>11110 11111</td>
<td>8</td>
</tr>
</tbody>
</table>

Creator: Malik Magdon-Ismail
Counting: 9 / 17
Goody Bags →
Build-Up Counting

Number of binary sequences of length $n$: $2^n$.

$\binom{n}{k} = \text{number binary sequences of length } n \text{ with exactly } k \text{ 1's} \quad 0 \leq k \leq n.$

Length-3 sequences:

000 001 010 011 100 101 110 111

Length-4 sequences:

0000 0001 0010 0011 0100 0101 0110 0111
1000 1001 1010 1011 1100 1101 1110 1111

Length-5 sequences:

00000 00001 00010 00011 00100 00101 00110 00111
01000 01001 01010 01011 01100 01101 01110 01111
10000 10001 10010 10011 10100 10101 10110 10111
11000 11001 11010 11011 11100 11101 11110 11111

\[
\begin{array}{c|cccccccc}
\binom{n}{k} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
0 & 1 \\
1 & 1 & 1 \\
2 & 1 & 2 & 1 \\
3 & 1 & 3 & 3 & 1 \\
4 & 1 & 4 & 6 & 4 & 1 \\
5 & 1 & 5 & 10 & 10 & 5 & 1 \\
6 & 1 \\
7 & 1 \\
8 & 1 \\
\end{array}
\]
Build-Up Counting

Number of binary sequences of length $n$: $2^n$.

$\binom{n}{k}$ = number binary sequences of length $n$ with exactly $k$ 1's \hspace{1cm} 0 \leq k \leq n.$

<table>
<thead>
<tr>
<th>$k$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\{n\text{-sequence with } k \text{ 1's}\} = 0 \cdot \binom{n-1}{k} \text{-sequence with } k \text{ 1's} \cup 1 \cdot \binom{n-1}{k-1} \text{-sequence with } (k-1) \text{ 1's}$
Build-Up Counting

Number of binary sequences of length $n$: $2^n$.

$$\binom{n}{k} = \text{number binary sequences of length } n \text{ with exactly } k \text{ 1's } \quad 0 \leq k \leq n.$$ 

<table>
<thead>
<tr>
<th>$n$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>6</td>
<td>15</td>
<td>20</td>
<td>15</td>
<td>6</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\{n\text{-sequence with } k \text{ 1's}\} = 0 \cdot \binom{(n-1)}{k} \text{-sequence with } k \text{ 1's} \cup 1 \cdot \binom{(n-1)}{k-1} \text{-sequence with } (k-1) \text{ 1's}$
Build-Up Counting

Number of binary sequences of length $n$: $2^n$.

$\binom{n}{k} = \text{number binary sequences of length } n \text{ with exactly } k \text{ 1's} \quad 0 \leq k \leq n.$

**Length-3 sequences:**
000 001 010 011 100 101 110 111

**Length-4 sequences:**
0000 0001 0010 0011 0100 0101 0110 0111
1000 1001 1010 1011 1100 1101 1110 1111

**Length-5 sequences:**
00000 00001 00010 00011 00100 00101 00110 00111
01000 01001 01010 01011 01100 01101 01110 01111
10000 10001 10010 10011 10100 10101 10110 10111
11000 11001 11010 11011 11100 11101 11110 11111

<table>
<thead>
<tr>
<th>$\binom{n}{k}$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>6</td>
<td>15</td>
<td>20</td>
<td>15</td>
<td>6</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\{n\text{-sequence with } k \text{ 1's}\} = 0 \cdot \left\{ (n-1)\text{-sequence with } k \text{ 1's} \right\} \cup 1 \cdot \left\{ (n-1)\text{-sequence with } (k-1) \text{ 1's} \right\}
\]

\[
\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \quad \text{(sum rule)}
\]

base cases: $\binom{n}{0} = 1; \binom{n}{n} = 1.$
Build-Up Counting

Number of binary sequences of length $n$: $2^n$.

\[
\binom{n}{k} = \text{number binary sequences of length } n \text{ with exactly } k \text{ 1's } \quad 0 \leq k \leq n.
\]

Length-3 sequences:
000 001 010 011 100 101 110 111

Length-4 sequences:
0000 0001 0010 0011 0100 0101 0110 0111
1000 1001 1010 1011 1100 1101 1110 1111

Length-5 sequences:
00000 00001 00010 00011 00100 00101 00110 00111
01000 01001 01010 01011 01100 01101 01110 01111
01000 10001 10010 10011 10100 10101 10110 10111
10001 10010 10011 10100 10101 10110 10111 11000
11001 11010 11011 11100 11101 11110 11111

\[
\begin{array}{c|cccccccc}
\binom{n}{k} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
0 & 1 & & & & & & & & \\
1 & 1 & 1 & & & & & & & \\
2 & 1 & 2 & 1 & & & & & & \\
3 & 1 & 3 & 3 & 1 & & & & & \\
4 & 1 & 4 & 6 & 4 & 1 & & & & \\
5 & 1 & 5 & 10 & 10 & 5 & 1 & & & \\
6 & 1 & 6 & 15 & 20 & 15 & 6 & 1 & & \\
7 & 1 & 7 & 21 & 35 & 35 & 21 & 7 & 1 & \\
8 & 1 & 8 & 28 & 56 & 70 & 56 & 28 & 8 & 1 \\
\end{array}
\]

Pascal’s Triangle

\[
\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \quad \text{(sum rule)}
\]

base cases: $\binom{n}{0} = 1; \binom{n}{n} = 1.$
$Q(n, k) =$ number of goody-bags of $n$ candies with $k$ colors
Build-up Counting for Goody Bags

\[ Q(n, k) = \text{number of goody-bags of } n \text{ candies with } k \text{ colors} \]

Build-up counting: there are \((n + 1)\) types of goody-bag.

0 1 2 3 \(\cdots\) \(n\)
Build-up Counting for Goody Bags

\[ Q(n, k) = \text{number of goody-bags of } n \text{ candies with } k \text{ colors} \]

Build-up counting: there are \((n + 1)\) types of goody-bag.

\[
\begin{align*}
Q(n, k - 1) & \quad Q(n - 1, k - 1) & \quad Q(n - 2, k - 1) & \quad Q(n - 3, k - 1) & \cdots & \quad Q(0, k - 1) \\
0 & \quad 1 & \quad 2 & \quad 3 & \cdots & \quad n
\end{align*}
\]
Build-up Counting for Goody Bags

\[ Q(n, k) = \text{number of goody-bags of } n \text{ candies with } k \text{ colors} \]

Build-up counting: there are \((n + 1)\) types of goody-bag.

\[
\begin{align*}
Q(n, k - 1) & \quad Q(n - 1, k - 1) & \quad Q(n - 2, k - 1) & \quad Q(n - 3, k - 1) & \cdots & Q(0, k - 1) \\
\end{align*}
\]

\[ Q(n, k) = Q(0, k - 1) + Q(1, k - 1) + \cdots + Q(n, k - 1) \quad \text{(sum rule)} \]
Build-up Counting for Goody Bags

\[ Q(n, k) = \text{number of goody-bags of } n \text{ candies with } k \text{ colors} \]

Build-up counting: there are \((n + 1)\) types of goody-bag.

\[
\begin{align*}
Q(n, k - 1) & \quad Q(n - 1, k - 1) & \quad Q(n - 2, k - 1) & \quad Q(n - 3, k - 1) & \quad \cdots & \quad Q(0, k - 1) \\
Q(n, k) = Q(0, k - 1) + Q(1, k - 1) + \cdots + Q(n, k - 1) & \quad \text{(sum rule)}
\end{align*}
\]

<table>
<thead>
<tr>
<th>(Q(n, k))</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n = 0)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(n = 1)</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n = 2)</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n = 3)</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n = 4)</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n = 5)</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(Q(n, 1) = 1; Q(0, k) = 1\)
Build-up Counting for Goody Bags

\[ Q(n, k) = \text{number of goody-bags of } n \text{ candies with } k \text{ colors} \]

Build-up counting: there are \((n+1)\) types of goody-bag.

\[
Q(n, k - 1) \quad Q(n - 1, k - 1) \quad Q(n - 2, k - 1) \quad Q(n - 3, k - 1) \quad \cdots \quad Q(0, k - 1)
\]

\[
Q(n, k) = Q(0, k - 1) + Q(1, k - 1) + \cdots + Q(n, k - 1)
\]  
  \text{(sum rule)}

<table>
<thead>
<tr>
<th>(Q(n, k))</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>6</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(Q(n, 1) = 1; \ Q(0, k) = 1\)
Build-up Counting for Goody Bags

\[ Q(n, k) = \text{number of goody-bags of } n \text{ candies with } k \text{ colors} \]

Build-up counting: there are \((n + 1)\) types of goody-bag.

\[
\begin{align*}
Q(n, k - 1) & \quad Q(n - 1, k - 1) & \quad Q(n - 2, k - 1) & \quad Q(n - 3, k - 1) \quad \cdots \quad Q(0, k - 1) \\
\end{align*}
\]

\[ Q(n, k) = Q(0, k - 1) + Q(1, k - 1) + \cdots + Q(n, k - 1) \quad \text{(sum rule)} \]

<table>
<thead>
<tr>
<th>(Q(n, k))</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n=0)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(n=1)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>(n=2)</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>15</td>
<td>21</td>
<td>28</td>
<td>36</td>
<td>45</td>
<td>55</td>
<td>66</td>
</tr>
<tr>
<td>(n=3)</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>20</td>
<td>35</td>
<td>56</td>
<td>84</td>
<td>120</td>
<td>165</td>
<td>220</td>
<td>286</td>
</tr>
<tr>
<td>(n=4)</td>
<td>1</td>
<td>5</td>
<td>15</td>
<td>35</td>
<td>70</td>
<td>126</td>
<td>210</td>
<td>330</td>
<td>495</td>
<td>715</td>
<td>1001</td>
</tr>
<tr>
<td>(n=5)</td>
<td>1</td>
<td>6</td>
<td>21</td>
<td>56</td>
<td>126</td>
<td>252</td>
<td>462</td>
<td>792</td>
<td>1287</td>
<td>2002</td>
<td>3003</td>
</tr>
</tbody>
</table>

\(Q(n, 1) = 1\); \(Q(0, k) = 1\)
Build-up Counting for Goody Bags

\[ Q(n, k) = \text{number of goody-bags of } n \text{ candies with } k \text{ colors} \]

Build-up counting: there are \((n + 1)\) types of goody-bag.

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & \cdots & n \\
Q(n, k - 1) & Q(n - 1, k - 1) & Q(n - 2, k - 1) & Q(n - 3, k - 1) & \cdots & Q(0, k - 1)
\end{array}
\]

\[ Q(n, k) = Q(0, k - 1) + Q(1, k - 1) + \cdots + Q(n, k - 1) \quad \text{(sum rule)} \]

\[
\begin{array}{cccccccc}
\begin{array}{cccccccc}
Q(n, k) & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
2 & 1 & 3 & 6 & 10 & 15 & 21 & 28 & 36 & 45 & 55 & 66 \\
3 & 1 & 4 & 10 & 20 & 35 & 56 & 84 & 120 & 165 & 220 & 286 \\
4 & 1 & 5 & 15 & 35 & 70 & 126 & 210 & 330 & 495 & 715 & 1001 \\
5 & 1 & 6 & 21 & 56 & 126 & 252 & 462 & 792 & 1287 & 2002 & 3003
\end{array}
\end{array}
\]

Challenge problems we had earlier.

1. (5 candies, 10 colors) \(\rightarrow\) 2002 goody-bags.

2. How many 5 goody-bag bulk packs (goody-bags have 3 candies of 3 colors)?
   There are 10 types of goody-bag; 5 in a bulk pack. So we need \(Q(5, 10) = 2002\).
10 goody-bags with 3 candies of 3 colors. Can label those goody bags using \{1, 2, \ldots, 10\}.

\[
\begin{array}{cccccccccc}
\{\text{red} \text{ red} \text{ red}}\} & \{\text{red} \text{ red} \text{ blue}}\} & \{\text{red} \text{ red} \text{ green}}\} & \{\text{red} \text{ blue} \text{ blue}}\} & \{\text{red} \text{ blue} \text{ green}}\} & \{\text{red} \text{ green} \text{ green}}\} & \{\text{blue} \text{ blue} \text{ green}}\} & \{\text{blue} \text{ green} \text{ green}}\} & \{\text{green} \text{ green} \text{ green}}\} \\
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10
\end{array}
\]

1-to-1 correspondence between goody-bags and the set \{1, 2, \ldots, 10\}, a bijection.
Counting One Set By Counting Another: Bijection

10 goody-bags with 3 candies of 3 colors. Can label those goody bags using \{1, 2, \ldots, 10\}.

\[
\begin{array}{cccccccccc}
\{ & \text{red} & \} & \{ & \text{red} & \text{blue} & \} & \{ & \text{red} & \text{green} & \} & \{ & \text{blue} & \text{green} & \} & \{ & \text{green} & \} & \{ & \text{red} & \} & \{ & \text{red} & \text{green} & \} & \{ & \text{blue} & \text{green} & \} & \{ & \text{green} & \} \\
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\end{array}
\]

1-to-1 correspondence between goody-bags and the set \{1, 2, \ldots, 10\}, a bijection.

\[
\begin{array}{cccccc}
A & B \\
\text{not a function} & \\
\end{array}
\quad
\begin{array}{cccccc}
A & B \\
1\text{-to-1; not onto} \quad \text{(injection, } A \xrightarrow{\text{inj}} B) \quad |A| \leq |B| \\
\end{array}
\quad
\begin{array}{cccccc}
A & B \\
onto \text{; not 1-to-1} \quad \text{(surjection, } A \xrightarrow{\text{sur}} B) \quad |A| \geq |B| \\
\end{array}
\quad
\begin{array}{cccccc}
A & B \\
1\text{-to-1 and onto} \quad \text{(bijection, } A \xrightarrow{\text{bij}} B) \quad |A| = |B| \\
\end{array}
\]
Counting One Set By Counting Another: Bijection

10 goody-bags with 3 candies of 3 colors. Can label those goody bags using \{1, 2, \ldots, 10\}.

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\end{array}
\]

1-to-1 correspondence between goody-bags and the set \{1, 2, \ldots, 10\}, a bijection.

\[
A \bijection B \implies |A| = |B|. \text{ Can count } A \text{ by counting } B.
\]

Count menus by counting sequences \{BLD\}. Works because
- Every sequence specifies a distinct menu (1-to-1 mapping).
- Every menu corresponds to a sequence (the mapping is onto).
3 colors, red, blue, green. Consider the 7-candy goody-bag \{2\red, 3\blue, 2\green\}:
3 colors, $\bullet$, $\bullet$, $\bullet$. Consider the 7-candy goody-bag $\{2\bullet, 3\bullet, 2\bullet\}$:

<table>
<thead>
<tr>
<th>red candies</th>
<th>delimiter</th>
<th>blue candies</th>
<th>delimiter</th>
<th>green candies</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bullet \bullet$</td>
<td>$</td>
<td>$</td>
<td>$\bullet \bullet \bullet$</td>
<td>$</td>
</tr>
</tbody>
</table>

$\leftrightarrow$ infer color from position

$\bullet \bullet \mid \bullet \bullet \bullet \mid \bullet \bullet$

goody-bags $\mapsto$ 9-bit binary sequences with two 1’s.
3 colors, $\bullet$, $\bullet$, $\bullet$. Consider the 7-candy goody-bag $\{2\bullet, 3\bullet, 2\bullet\}$:

\[
\begin{array}{c|c|c}
\text{red candies} & \text{delimiter} & \text{blue candies} & \text{delimiter} & \text{green candies} \\
\bullet & | & \bullet \bullet & | & \bullet  \\
\end{array}
\quad \leftrightarrow \quad
\begin{array}{c|c|c|c|c|c|c}
\text{infer color from position} \\
\bullet & | & \bullet & | & \bullet & | & \bullet \\
\end{array}
\]

\text{goody-bags} \xrightarrow{\text{bij}} 9\text{-bit binary sequences with two 1’s.}

**Examples.**

\[
\begin{align*}
00100010101000 & \rightarrow \; \bullet \bullet | \bullet \bullet \bullet | \bullet | \bullet | \bullet \bullet \bullet \leftrightarrow \{2\bullet, 3\bullet, 1\bullet, 1\bullet, 3\bullet\} \\
1000011010000 & \rightarrow \; | \bullet \bullet \bullet \bullet | \bullet | \bullet \bullet \bullet \bullet \bullet \leftrightarrow \{0\bullet, 4\bullet, 0\bullet, 1\bullet, 4\bullet\}
\end{align*}
\]
Goody Bags Using Bijection to Binary Sequences

3 colors, ⬤, ⬤, ⬤. Consider the 7-candy goody-bag \{2⬸, 3⬸, 2⬸\}:

\[
\text{red candies} \quad \text{delimiter} \quad \text{blue candies} \quad \text{delimiter} \quad \text{green candies} \quad \leftrightarrow \quad \text{infer color from position}
\]

\[
\text{goody-bags} \quad \leftrightarrow \quad 9\text{-bit binary sequences with two 1’s.}
\]

Examples.

\[
00100010101000 \rightarrow \circ\circ | \circ\circ\circ | \circ | \circ | \circ\circ\circ \leftrightarrow \{2⬸, 3⬸, 1⬸, 1⬸, 3⬸\}
\]
\[
1000011010000 \rightarrow \circ\circ\circ\circ\circ | \circ | \circ\circ\circ\circ \leftrightarrow \{0⬸, 4⬸, 0⬸, 1⬸, 4⬸\}
\]

\(n\) candies and \(k\) colors \(\rightarrow\) \((k - 1)\) delimiters.

number of goody-bags with \(n\) candies of \(k\) colors = number of \((n + k - 1)\)-bit sequences with \((k - 1)\) 1’s
3 colors, •, •, •. Consider the 7-candy goody-bag \{2\bullet, 3\bullet, 2\bullet\}:

\[
\begin{array}{ccc}
\text{red candies} & | & \text{blue candies} & | & \text{green candies}
\end{array}
\leftrightarrow
\begin{array}{c}
\text{infer color from position}
\end{array}
\]

\[
\begin{array}{c|c|c}
\bullet \bullet & | & \bullet \bullet \bullet & | & \bullet \\
\end{array}
\leftrightarrow
\begin{array}{c|c|c}
\circ \circ & | & \circ \circ \circ & | & \circ \circ
\end{array}
\]

goody-bags \leftrightarrow 9-bit binary sequences with two 1’s.

**Examples.**

\[
\begin{align*}
00100010101000 & \rightarrow \circ \circ | \circ \circ \circ | \circ | \circ | \circ \circ \circ & \leftrightarrow & \{2\bullet, 3\bullet, 1\bullet, 1\bullet, 3\bullet\} \\
1000011010000 & \rightarrow | \circ \circ \circ \circ | \circ | \circ \circ \circ \circ & \leftrightarrow & \{0\bullet, 4\bullet, 0\bullet, 1\bullet, 4\bullet\}
\end{align*}
\]

\(n\) candies and \(k\) colors \(\rightarrow (k - 1)\) delimiters.

number of goody-bags with \(n\) candies of \(k\) colors = number of \((n + k - 1)\)-bit sequences with \((k - 1)\) 1’s

\[
Q(n, k) = \binom{n + k - 1}{k - 1}.
\]

\(\binom{n}{k}\) keeps popping up but we don’t have a formula for it.
Nonaligned King-Queen Positions on a Chessboard
Nonaligned King-Queen Positions on a Chessboard

<table>
<thead>
<tr>
<th>complex object</th>
<th>sequence to reconstruct it</th>
</tr>
</thead>
<tbody>
<tr>
<td>king-queen position</td>
<td>$c3g4$</td>
</tr>
<tr>
<td>number of positions</td>
<td>number of sequences $c_K r_K c_Q r_Q$</td>
</tr>
</tbody>
</table>

Every position gives a sequence with $r_k \neq r_Q$ and $c_K \neq c_Q$
Every such sequence is a unique valid position.
Nonaligned King-Queen Positions on a Chessboard

<table>
<thead>
<tr>
<th>complex object</th>
<th>sequence to reconstruct it</th>
</tr>
</thead>
<tbody>
<tr>
<td>king-queen position</td>
<td>c3g4</td>
</tr>
<tr>
<td>number of positions</td>
<td>number of sequences $c_K r_K c_Q r_Q$</td>
</tr>
</tbody>
</table>

Every position gives a sequence with $r_k \neq r_Q$ and $c_K \neq c_Q$
Every such sequence is a unique valid position.

Count sequences with $r_k \neq r_Q$ and $c_K \neq c_Q$. 
Nonaligned King-Queen Positions on a Chessboard

<table>
<thead>
<tr>
<th>complex object</th>
<th>sequence to reconstruct it</th>
</tr>
</thead>
<tbody>
<tr>
<td>king-queen position</td>
<td>c3g4</td>
</tr>
<tr>
<td>number of positions</td>
<td>number of sequences $c_K r_K c_Q r_Q$</td>
</tr>
</tbody>
</table>

Every position gives a sequence with $r_k \neq r_Q$ and $c_K \neq c_Q$.
Every such sequence is a unique valid position.

Count sequences with $r_k \neq r_Q$ and $c_K \neq c_Q$.

8 choices for $c_K$ and $r_K$; after choosing $c_K, r_K$, there are only 7 choices for $c_Q$ and $r_Q$.

By the product rule, the number of sequences is $8 \times 8 \times 7 \times 7 = 3136$.
By bijection counting, number of King-Queen positions is 3136.
Nonaligned Castle-Castle Positions on a Chessboard
### Nonaligned Castle-Castle Positions on a Chessboard

<table>
<thead>
<tr>
<th>complex object</th>
<th>sequence to reconstruct it</th>
</tr>
</thead>
<tbody>
<tr>
<td>castle-castle position</td>
<td>$c3g4$</td>
</tr>
<tr>
<td>number of positions</td>
<td>number of sequences $c_1r_1c_2r_2$</td>
</tr>
</tbody>
</table>

```
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |
| c3 | g4 |   |   |   |   |   |   |
| r1 |   |   |   |   |   |   |   |
| r2 |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
```

Creator: Malik Magdon-Ismail
Counting: 14 / 17
Permutations and Combinations →
Nonaligned Castle-Castle Positions on a Chessboard

<table>
<thead>
<tr>
<th>complex object</th>
<th>sequence to reconstruct it</th>
</tr>
</thead>
<tbody>
<tr>
<td>castle-castle position</td>
<td>(c3g4)</td>
</tr>
<tr>
<td>number of positions</td>
<td>number of sequences (c_1r_1c_2r_2)</td>
</tr>
</tbody>
</table>

**Problem.** \(c3g4\) and \(g4c3\) are the same position. Position ↔ *two* sequences with \(r_1 \neq r_2\) and \(c_1 \neq c_2\)

Twice as many sequences as positions!
Nonaligned Castle-Castle Positions on a Chessboard

<table>
<thead>
<tr>
<th>complex object</th>
<th>sequence to reconstruct it</th>
</tr>
</thead>
<tbody>
<tr>
<td>castle-castle position</td>
<td>c3g4</td>
</tr>
<tr>
<td>number of positions</td>
<td>number of sequences $c_1r_1c_2r_2$</td>
</tr>
</tbody>
</table>

**Problem.** $c3g4$ and $g4c3$ are the same position. 

position $\leftrightarrow$ two sequences with $r_1 \neq r_2$ and $c_1 \neq c_2$

Twice as many sequences as positions!

**Multiplicity Rule.** If each object in $A$ corresponds to $k$ objects in $B$, then $|B| = k|A|$.
Nonaligned Castle-Castle Positions on a Chessboard

<table>
<thead>
<tr>
<th>complex object</th>
<th>sequence to reconstruct it</th>
</tr>
</thead>
<tbody>
<tr>
<td>castle-castle position</td>
<td>c3g4</td>
</tr>
<tr>
<td>number of positions</td>
<td>number of sequences c1r1c2r2</td>
</tr>
</tbody>
</table>

**Problem.** c3g4 and g4c3 are the same position. position ↔ two sequences with \( r_1 \neq r_2 \) and \( c_1 \neq c_2 \)

Twice as many sequences as positions!

**Multiplicity Rule.** If each object in \( A \) corresponds to \( k \) objects in \( B \), then \( |B| = k|A| \).

8 choices for \( c_1 \) and \( r_1 \); after choosing \( c_1, r_1 \), there are only 7 choices for \( c_2 \) and \( r_2 \).

By the product rule, the number of sequences is \( 8 \times 8 \times 7 \times 7 = 3136 \).
By the multiplicity rule, number of Castle-Castle positions is \( \frac{1}{2} \times 3136 = 1568 \).
Permutations and Combinations

\[ S = \{1, 2, 3, 4\}, \]

the 2-orderings are:
\[ \{1 2, 1 3, 1 4, 2 1, 2 3, 2 4, 3 1, 3 2, 3 4, 4 1, 4 2, 4 3\} \]

(permutations)

the 2-subsets are:
\[ \{1 2, 1 3, 1 4, 2 3, 2 4, 3 4\} \]

(combinations)
Permutations and Combinations

\[ S = \{1, 2, 3, 4\}, \]

the 2-orderings are:
\[ \{1 \ 2, 1 \ 3, 1 \ 4, 2 \ 1, 2 \ 3, 2 \ 4, 3 \ 1, 3 \ 2, 3 \ 4, 4 \ 1, 4 \ 2, 4 \ 3\} \]

(permutations)

the 2-subsets are:
\[ \{1 \ 2, 1 \ 3, 1 \ 4, 2 \ 3, 2 \ 4, 3 \ 4\} \]

(combinations)

With \( n \) elements, by the product rule, the number of \( k \)-orderings is

\[
\text{number of } k\text{-orderings} = n \times (n - 1) \times (n - 2) \times \cdots \times (n - (k - 1)) = \frac{n!}{(n - k)!}.
\]

e.g. number of top-3 finishes in 10-person race is \( 10 \times 9 \times 8 = 10!/7! \).
Permutations and Combinations

\[ S = \{1, 2, 3, 4\}, \]

the 2-orderings are:
\[ \{1\, 2, \ 1\, 3, \ 1\, 4, \ 2\, 1, \ 2\, 3, \ 2\, 4, \ 3\, 1, \ 3\, 2, \ 3\, 4, \ 4\, 1, \ 4\, 2, \ 4\, 3\} \]

\((\text{permutations})\)

the 2-subsets are:
\[ \{1\, 2, \ 1\, 3, \ 1\, 4, \ 2\, 3, \ 2\, 4, \ 3\, 4\} \]

\((\text{combinations})\)

With \(n\) elements, by the product rule, the number of \(k\)-orderings is

\[
\text{number of } k\text{-orderings} = n \times (n - 1) \times (n - 2) \times \cdots \times (n - (k - 1)) = \frac{n!}{(n - k)!}.
\]

e.g. number of top-3 finishes in 10-person race is \(10 \times 9 \times 8 = 10!/7!\).

Pick a \(k\)-subset \((\binom{n}{k} \text{ ways})\) and reorder it in \(k!\) ways to get a \(k\)-ordering.

\[
\text{number of } k\text{-orderings} = \binom{n}{k} \times k! \\
\text{← product rule} \\
\text{← bijection to sequences with } k \text{'s}
\]
Permutations and Combinations

\[ S = \{1, 2, 3, 4\}, \]

the 2-orderings are:
\[
\{12, 13, 14, 21, 23, 24, 31, 32, 34, 41, 42, 43\}
\]

(permutations)

the 2-subsets are:
\[
\{12, 13, 14, 23, 24, 34\}
\]

(combinations)

With \( n \) elements, by the product rule, the number of \( k \)-orderings is

\[
\text{number of } k\text{-orderings} = n \times (n-1) \times (n-2) \times \cdots \times (n-(k-1)) = \frac{n!}{(n-k)!}.
\]

e.g. number of top-3 finishes in 10-person race is \( 10 \times 9 \times 8 = 10!/7! \).

Pick a \( k \)-subset (\( \binom{n}{k} \) ways) and reorder it in \( k! \) ways to get a \( k \)-ordering.

\[
\text{number of } k\text{-orderings} = \text{number of } k\text{-subsets} \times k! \\
= \binom{n}{k} \times k!
\]

← product rule

← bijection to sequences with \( k \) 1’s

\[
\text{number of } k\text{-subsets} = \binom{n}{k} = \frac{n!}{k!(n-k)!}
\]

Exercise. How many 10-bit binary sequences with four 1’s?
Binomial Theorem: \((x + y)^n = \sum_{i=1}^{n} \binom{n}{i} x^i y^{n-i}\)

\[(x + y)^3 = (x + y)(x + y)(x + y) = xxx + xxy + xyx + xyy + yxx + yxy + yyx + yyy = x^3 + 3x^2y + 3xy^2 + y^3.\]

(All length-3 binary sequences \(b_1b_2b_3\) where each \(b_i \in \{x, y\}\))
Binomial Theorem: \((x + y)^n = \sum_{i=1}^{n} \binom{n}{i} x^i y^{n-i}\)

\((x + y)^3 = (x + y)(x + y)(x + y) = x^3 + 3x^2y + 3xy^2 + y^3.\)

(All length-3 binary sequences \(b_1b_2b_3\) where each \(b_i \in \{x, y\}\))

\((x + y)^n = x^n + (\_\_\_)x^{n-1}y + (\_\_\_)x^{n-2}y^2 + (\_\_\_)x^{n-3}y^3 + \cdots + (\_\_\_)xy^{n-1} + y^n\)

\(\uparrow\) strings with \((n-1)\) \(x\)'s \hspace{1cm} \(\uparrow\) strings with \((n-2)\) \(x\)'s \hspace{1cm} \(\uparrow\) strings with \((n-3)\) \(x\)'s \hspace{1cm} \(\uparrow\) strings with 1 \(x\)
Binomial Theorem: $(x + y)^n = \sum_{i=1}^{n} \binom{n}{i} x^i y^{n-i}$

$(x + y)^3 = (x + y)(x + y)(x + y) = xxx + xxy + xyx + yyy + yxx + yxy + yyy + yyy$

$= x^3 + 3x^2y + 3xy^2 + y^3.$

(All length-3 binary sequences $b_1b_2b_3$ where each $b_i \in \{x, y\}$)

$(x + y)^n = x^n + (\_\_\_\_)x^{n-1}y + (\_\_\_\_)x^{n-2}y^2 + (\_\_\_\_)x^{n-3}y^3 + \cdots + (\_\_\_\_)xy^{n-1} + y^n$

\[\begin{array}{cccc}
\uparrow & \uparrow & \uparrow & \uparrow \\
\text{strings with} & \text{strings with} & \text{strings with} & \text{strings with} \\
(n-1) \text{ } x \text{'s} & (n-2) \text{ } x \text{'s} & (n-3) \text{ } x \text{'s} & 1 \text{ } x \\
\binom{n}{n-1} & \binom{n}{n-2} & \binom{n}{n-3} & \binom{n}{1}
\end{array}\]
Binomial Theorem: \((x + y)^n = \sum_{i=1}^{n} \binom{n}{i} x^i y^{n-i}\)

\[(x + y)^3 = (x + y)(x + y)(x + y) = x^3 + 3x^2y + 3xy^2 + y^3.\]

(All length-3 binary sequences \(b_1b_2b_3\) where each \(b_i \in \{x, y\}\))

\[(x + y)^n = x^n + \binom{n}{n-1}x^{n-1}y + \binom{n}{n-2}x^{n-2}y^2 + \binom{n}{n-3}x^{n-3}y^3 + \cdots + \binom{n}{1}xy^{n-1} + y^n\]

\[= \binom{n}{n}x^n + \binom{n}{n-1}x^{n-1}y + \binom{n}{n-2}x^{n-2}y^2 + \binom{n}{n-3}x^{n-3}y^3 + \cdots + \binom{n}{1}xy^{n-1} + \binom{n}{0}y^n\]
Binomial Theorem: \((x + y)^n = \sum_{i=1}^{n} \binom{n}{i} x^i y^{n-i}\)

\[(x + y)^3 = (x + y)(x + y)(x + y) = xxx + xxy + xyx + yyy + yxx + yxy + yyx + yyy = x^3 + 3x^2y + 3xy^2 + y^3.\]

(All length-3 binary sequences \(b_1b_2b_3\) where each \(b_i \in \{x, y\}\))

\[(x + y)^n = x^n + (\binom{n}{n-1})x^{n-1}y + (\binom{n}{n-2})x^{n-2}y^2 + (\binom{n}{n-3})x^{n-3}y^3 + \cdots + (\binom{n}{1})xy^{n-1} + y^n\]

Example. What is the coefficient of \(x^7\) in the expansion of \((\sqrt{x} + 2x)^{10}\)

Need \((\sqrt{x})^i(2x)^{10-i} \sim x^7\), which implies \(i = 6\).

The \(x^7\) term is \(\binom{10}{6}(\sqrt{x})^6(2x)^4\)

Coefficient of \(x^7\) is \(\binom{10}{6} \times 2^4 = 3360.\)
To count complex objects, give a sequence of “instructions” that can be used to construct a complex object.
To count complex objects, give a sequence of “instructions” that can be used to construct a complex object.

- *Every* sequence of instructions gives a *unique* complex object.
To count complex objects, give a sequence of “instructions” that can be used to construct a complex object.

- *Every* sequence of instructions gives a *unique* complex object.
- There is a sequence of instructions for *every* complex object.
To count complex objects, give a sequence of “instructions” that can be used to construct a complex object.

- *Every* sequence of instructions gives a *unique* complex object.
- There is a sequence of instructions for *every* complex object.

Count the number of possible *sequences* of instructions, which equals the number of complex objects.