Counting
  – Chapter 13
Overview

- Counting sequences.
- Build-up counting.
- Counting one set by counting another: bijection.
- Permutations and combinations.
Discrete Math is About Objects We Can Count

• Three colors of candy: red, blue, green
• A goody-bag has 3 candies. How many distinct goody-bags?
• (Only the number of each color matters: bags with different orderings are the “same” goody-bag.)

• Challenge Problems.
  – What if there are 5 candies per goody-bag and 10 colors of candy?
  – 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!
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,
  \(N = N_1 + N_2\)

• Going back to the binary example:

  \[|\{b_1b_2b_3\}| = |\{b_1b_2 \cdot 0\}| + |\{b_1b_2 \cdot 1\}| \quad \text{[sum rule]}\]
  \[= |\{b_1b_2\}| \times 2\]
  \[= (|\{b_1 \cdot 0\}| + |\{b_1 \cdot 1\}|) \times 2 \quad \text{[sum rule]}\]
  \[= |\{b_1\}| \times 2 \times 2\]
  \[= 2 \times 2 \times 2\]
Product Rule

- **Number of choices rule**
  \[ |\{b_1 b_2 b_3\}| = 2 \times 2 \times 2 \]

- **Product Rule.** 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.**
  – \( \text{breakfast} \in \{ \text{pancake, waffle, coffee} \} \)
  – \( \text{lunch} \in \{ \text{burger, coffee} \} \)
  – \( \text{dinner} \in \{ \text{salad, steak, coffee} \} \)
    \[ |\{\text{BLD}\}| = 3 \times 2 \times 3 = 18 \]
  – (every menu is a sequence \( \text{BLD} \) and every sequence \( \text{BLD} \) is a unique menu.)

• **NY Plates.**
  – A NY plate has the form \( \{ABC - 1234\} \)
    \[ |\{ABC - 1234\}| = 26 \times 26 \times 26 \times 10 \times 10 \times 10 \times 10 \approx 176M \]

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

• **Passwords.**
  – Use: \{a, \ldots, z\}, \{A, \ldots, Z\}, \{0, \ldots, 9\}, special: \{!, @, #, $, %, &, *, (, )\}
  – Rules: Length is 8. Must have at least one special.
  – Total number is the sum of valid and invalid (no special symbol) passwords
    \[
    |\{\text{passwords}\}| = 72 \times 72 \times \cdots \times 72 = 72^8 \quad \text{[product rule]}
    = |\{\text{valid}\}| + |\{\text{invalid}\}| \quad \text{[sum rule]}
    = |\{\text{valid}\}| + 62^8 \quad \text{[product rule]}
    |\{\text{valid}\}| = 72^8 - 62^8 \approx 5 \times 10^{14}
    \]
  – (1 millisecond to test $\rightarrow$ about 6 months on 32K cores.)

• **Committees.**
  – We have 10 students. How many ways to form a party planning committee?
  – Each student can be in or out of the committee:
    • e.g. \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\}
    • Then 1 1 0 1 0 0 0 0 1 0 $\leftrightarrow$ \{s_1, s_2, s_4, s_9\}, 0 0 0 0 0 0 0 0 0 0 $\leftrightarrow$ $\emptyset$
  – Aha: $|\{\text{committees}\}| = |\{10 - \text{bit binary strings}\}|$
    \[
    2 \times 2 \times \cdots \times 2 = 2^{10} = 1024
    \]
Build-up Counting

• Already saw that the total number of binary sequences of length $n$ is $2^n$.

• How about the number binary sequences of length $n$ with exactly $k$ 1’s, $0 \leq k \leq n$?
  
  – Denote this number by $\binom{n}{k}$.

• First, tinker!
  
  – Length-3 sequences:
    000, 001, 010, 011, 100, 101, 110, 111.

\[
\begin{array}{cccccccccc}
\binom{n}{k} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
0 & 1 &  &  &  &  &  &  &  &  \\
1 & 1 & 1 &  &  &  &  &  &  &  \\
2 & 1 & 2 & 1 &  &  &  &  &  &  \\
3 & 1 & 3 & 3 & 1 &  &  &  &  &  \\
4 &  &  &  &  &  &  &  &  &  \\
5 &  &  &  &  &  &  &  &  &  \\
6 &  &  &  &  &  &  &  &  &  \\
7 &  &  &  &  &  &  &  &  &  \\
8 &  &  &  &  &  &  &  &  &  \\
\end{array}
\]
Build-up Counting

• Already saw that the total number of binary sequences of length $n$ is $2^n$
• How about the number binary sequences of length $n$ with exactly $k$ 1’s, $0 \leq k \leq n$?
  
  – Denote this number by $\binom{n}{k}$

• First, tinker!
  
  – 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

\[
\begin{array}{cccccccccc}
\binom{n}{k} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
0 & 1 & & & & & & & & \\
1 & 1 & 1 & & & & & & & \\
2 & 1 & 2 & 1 & & & & & & \\
3 & 1 & 3 & 3 & 1 & & & & & \\
4 & 1 & 4 & 6 & 4 & 1 & & & & \\
5 & & & & & & & & & \\
6 & & & & & & & & & \\
7 & & & & & & & & & \\
8 & & & & & & & & & \\
\end{array}
\]
Build-up Counting

- Already saw that the total number of binary sequences of length $n$ is $2^n$
- How about the number binary sequences of length $n$ with exactly $k$ 1’s, $0 \leq k \leq n$?
  - Denote this number by $\binom{n}{k}$

First, tinker!

- 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

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

Pascal’s Triangles!
Build-up Counting

- Already saw that the total number of binary sequences of length $n$ is $2^n$
- How about the number binary sequences of length $n$ with exactly $k$ 1’s, $0 \leq k \leq n$?
  - Denote this number by $\binom{n}{k}$
- First, tinker!
  - 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

Here is a table showing the values of $\binom{n}{k}$ for different values of $n$ and $k$:

<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></td>
<td>1</td>
<td>1</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></td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>6</td>
<td>15</td>
<td>20</td>
<td>15</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>1</td>
<td>7</td>
<td>21</td>
<td>35</td>
<td>35</td>
<td>21</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>1</td>
<td>8</td>
<td>28</td>
<td>56</td>
<td>70</td>
<td>56</td>
<td>28</td>
</tr>
</tbody>
</table>

Pascal’s Triangles!
Let’s try to come up with a formula:

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

– Hm, looks like induction!

Sum rule:

\[ \binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \]

Base cases:

\[ \binom{1}{0} = 1, \binom{1}{1} = 1 \]

– More generally, for any \( n \):

\[ \binom{n}{0} = 1, \binom{n}{n} = 1 \]

---

<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>1</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></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>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></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></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>7</td>
<td>21</td>
<td>35</td>
<td>35</td>
<td>21</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>8</td>
<td>28</td>
<td>56</td>
<td>70</td>
<td>56</td>
<td>28</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

Pascal’s Triangles!
Build-up Counting for Goody Bags

• Let $Q(n, k) =$ number of goody-bags of $n$ candies with $k$ colors

• First, tinker!
  – Suppose we have $n$ candies but only 1 color (red)
    $$Q(n, 1) = 1$$
  – Suppose we have zero candies and $k$ colors
    $$Q(0, k) = 1$$
  – Suppose we have 1 candy and $k$ colors
    $$Q(1, k) = k$$

• Build-up counting: there are $(n + 1)$ types of goody-bag:
  – Goody-bags that contain 0 red candies (and $k - 1$ other colors):
    $$Q(n, k - 1)$$
  – Goody-bags that contain exactly 1 red candy (so if I remove it, I have none):
    $$Q(n - 1, k - 1)$$
  – Goody-bags that contain exactly 2 red candies:
    $$Q(n - 2, k - 1)$$
  
  
  ...  

  $$Q(0, k - 1)$$
• I can express $Q(n, k)$ recursively as follows:
  
$$Q(n, k) = Q(n, k - 1) + Q(n - 1, k - 1) + \cdots + Q(0, k - 1)$$

• Let’s look at the recursive table

<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>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q(n, k)$</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>
<td>1</td>
</tr>
</tbody>
</table>
I can express $Q(n, k)$ recursively as follows:

$$Q(n, k) = Q(n, k - 1) + Q(n - 1, k - 1) + \cdots + Q(0, k - 1)$$

Let’s look at the recursive table:

<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></td>
<td></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></td>
<td></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></td>
<td></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></td>
<td></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></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>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>

$k$

$n$

- Build-up Counting for Goody Bags, cont’d

- I can express $Q(n, k)$ recursively as follows:
  
  $$Q(n, k) = Q(n, k - 1) + Q(n - 1, k - 1) + \cdots + Q(0, k - 1)$$

- Let’s look at the recursive table

<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></td>
<td></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></td>
<td></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></td>
<td></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></td>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</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>
• I can express $Q(n, k)$ recursively as follows:
  
  $$Q(n, k) = Q(n, k - 1) + Q(n - 1, k - 1) + \cdots + Q(0, k - 1)$$

• Let’s look at the recursive table

<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></td>
</tr>
<tr>
<td>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>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>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>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>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>

• What’s another way to group $Q(n, k)$?
  – All goody-bags that contain at least 1 red candy (already have a $k$ colors):
    $$Q(n - 1, k)$$
  – Plus all bags that have no red candies (have at most $k - 1$ colors):
    $$Q(n, k - 1)$$
• I can express $Q(n, k)$ recursively as follows:
  
  $$Q(n, k) = Q(n, k - 1) + Q(n - 1, k - 1) + \cdots + Q(0, k - 1)$$

• Let’s look at the recursive table

<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>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>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>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>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>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>

• Challenge problems we had earlier.
  – (5 candies, 10 colors) → 2002 goody-bags.
  – 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
    
    $Q(5, 10) = 2002$
**Counting One Set By Counting Another:**

- We saw that there are 10 goody-bags with 3 candies of 3 colors.
  - Can label those goody bags using \{1,2, ..., 10\}

\[
\begin{align*}
\{\text{\circ \circ \circ}\} & \quad \{\text{\circ \circ \bullet}\} & \quad \{\text{\bullet \circ \circ}\} & \quad \{\text{\circ \bullet \circ}\} & \quad \{\text{\circ \bullet \bullet}\} & \quad \{\text{\bullet \circ \bullet}\} & \quad \{\text{\bullet \bullet \circ}\} & \quad \{\text{\bullet \bullet \bullet}\} & \quad \{\text{\circ \circ \circ}\} & \quad \{\text{\circ \circ \circ}\} \\
\uparrow & \quad \uparrow & \quad \uparrow & \quad \uparrow & \quad \uparrow & \quad \uparrow & \quad \uparrow & \quad \uparrow & \quad \uparrow \\
1 & \quad 2 & \quad 3 & \quad 4 & \quad 5 & \quad 6 & \quad 7 & \quad 8 & \quad 9 & \quad 10
\end{align*}
\]

- There is a 1-1 correspondence between goody-bags and the set \{1,2, ..., 10\}
  - We call this a *bijection*!

- Some examples of bijections and other relations:

  
  1-1, but **not** onto.
  (injection, \(A \rightarrow B\))
  \(|A| \leq |B|\)

  onto; **not** 1-1
  (surjection, \(A \rightarrow B\))
  \(|A| \geq |B|\)

  onto and 1-1
  (bijection, \(A \rightarrow B\))
  \(|A| = |B|\)

  not a function
Counting One Set By Counting Another: Bijection, cont’d

- \( A \xrightarrow{Bij} B \) implies \(|A| = |B|\). Can count \( A \) 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).
Goody Bags Using Bijection to Binary Sequences

• Suppose we have 3 candy colors: red, green, blue
• Consider the 7-candy goody-bag: \{red, red, blue, blue, blue, green, green\}

<table>
<thead>
<tr>
<th>red candies</th>
<th>delimiter</th>
<th>blue candies</th>
<th>delimiter</th>
<th>green candies</th>
<th>infer color from position</th>
</tr>
</thead>
<tbody>
<tr>
<td>● ●</td>
<td></td>
<td>● ● ●</td>
<td></td>
<td>● ● ● ● ●</td>
<td>○ ○ ○ ○ ○ ○ ○ ○ ○ ○</td>
</tr>
</tbody>
</table>

  – Order all candies according to color: red first, then blue, then green
  – Color doesn’t matter anymore
  – Hm, this looks like a binary sequence:

    001000100

  – What is the binary sequence for bag \{red, red, red, green, green, green, green\}?

    | red candies | delimiter | green candies |
    |------------|-----------|---------------|
    | ● ● ●       |           | ● ● ● ● ●     |

    000110000

• Can represent all goody bags with 9 bits
• More examples.

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

• In general, if we have \(n\) candies and \(k\) colors, how many delimiters do we have?

\[
(k - 1)
\]

– i.e., number of goody-bags with \(n\) candies of \(k\) colors =

number of \((n + k - 1)\)-bit sequences with \((k - 1)\) 1’s

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

• This is called sampling with replacement

• Hm, \(\binom{n}{k}\) keeps popping up but we don’t have a formula for it.
• Consider the set \( S = \{1,2,3,4\} \)
  
  All 2-orderings of \( S \) are: \{12,13,14,21,23,24,31,32,34,41,42,43\}
  
  • Permutations: order matters
  
  All 2-subsets of \( S \) are: \{12,13,14,23,24,34\}
  
  • Combinations: order doesn’t matter

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

\[
\begin{align*}
  n \times (n-1) \times (n-2) \times \cdots \times (n-(k-1)) &= n! \\
  &= \frac{n!}{(n-k)!}
\end{align*}
\]

  
  – e.g. number of top-3 finishes in 10-person race is

\[
10 \times 9 \times 8 = \frac{10!}{7!}
\]
Permutations and Combinations, cont’d

- Consider the set \( S = \{1,2,3,4\} \)
  - All 2-orderings of \( S \) are: \{12,13,14,21,23,24,31,32,34,41,42,43\}
    - Permutations: order matters
  - All 2-subsets of \( S \) are: \{12,13,14,23,24,34\}
    - Combinations: order doesn’t matter

- Here’s another way to count all \( k \)-orderings
  - First, pick a \( k \)-subset, then re-order it in all possible ways
  - How many \( k \)-subsets are there?
    \[ \binom{n}{k} \]
  - How many ways can we re-order a set?
    \[ k \times (k - 1) \times \cdots \times 1 = k! \]

  number of \( k \)-orderings = number of \( k \)-subsets \( \times k! \) \[ \text{[product rule]} \]

  \[ = \binom{n}{k} \times k! \] \[ \text{[bijection to sequences with } k \text{ 1’s]} \]
• First method
\[
n \times (n - 1) \times (n - 2) \times \cdots \times (n - (k - 1)) = \frac{n!}{(n - k)!}
\]

• Second method
\[
\text{number of } k\text{-orderings} = \text{number of } k\text{-subsets} \times k! \quad \text{[product rule]}
\]
\[
\frac{n!}{(n-k)!} = \binom{n}{k} \times k! \quad \text{[bijection to sequences with } k \text{ 1's]}
\]

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

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

• We learn the formulas for small \(n\) in high school, e.g.,:
  
  \[(x + y)^3 = xxx + xxy + xyx + xyy + yxx + yxy + yyx + yyy
  = x^3 + 3x^2y + 3xy^2 + y^3\]

  – (All length-3 binary sequences \(b_1 b_2 b_3\) where each \(b_i \in \{x, y\}\))

• Of course, as we increase \(n\) the number of terms grows quickly, so we want a nice clean formula
  – The Binomial Theorem!

• In general, each term has combined power \(n\):
  
  \[(x + y)^n = x^n + (? )x^{n-1}y + (? )x^{n-2}y^2 + \cdots + (? )xy^{n-1} + y^n\]

  – How many strings with \((n - 1)\) \(x\)'s (first coefficient)?
    \[\binom{n}{n - 1}\]

  – How many strings with \((n - 2)\) \(x\)'s (second coefficient)?
    \[\binom{n}{n - 2}\]

• Finally,
  
  \[(x + y)^n = x^n + \binom{n}{n - 1} x^{n-1}y + \binom{n}{n - 2} x^{n-2}y^2 + \cdots + \binom{n}{1} xy^{n-1} + y^n\]
Binomial Theorem 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 \(\binom{10}{6} \times 2^4 = 3360\)
General Approach to Counting Complex Objects

• 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.