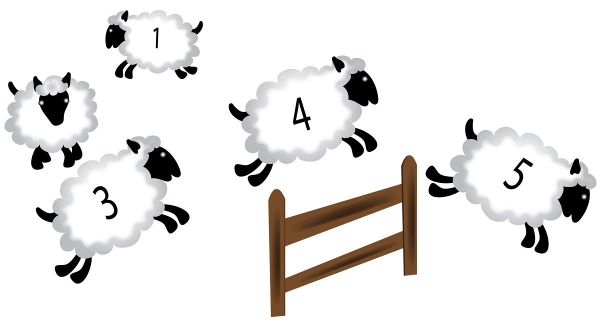
Foundations of Computer Science
Lecture 13

Counting



Last Time

1. Be careful of what you read in the media: sex in America.
2. Bipartite Graphs and Matching (Hall's Theorem).
3. Stable Marriage.
4. Conflict Graphs and Coloring.
   - Every tree is 2-colorable.
5. Other Graph Problems
   - Connected components, spanning tree, Euler paths, network flow. ([EASY])
   - Hamiltonian paths, K-center, vertex cover, dominating set. ([HARD])

Today: Counting

1. Counting sequences.
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.

Discrete Math Is About Objects We Can Count

Three colors of candy:

🔴, 🔵, 🟢.

A goody-bag has 3 candies. How many distinct goody-bags?
(Only the number of each color matters: {🔴🔴🟢} and {🔴🟢🔴} are the "same" goody-bag.)

{🔴🔴🔴} {🔴🔴🔵} {🔴🔴🟢} {🔴🔵🔵} {🔴🔵🟢} {🔴🟢🟢} {🔵🔵🔵} {🔵🔵🟢} {🔵🟢🟢} {🟢🟢🟢}

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!

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,

$$\{b_1b_2b_3\} = \{b_1b_2\bullet 0\} \cup \{b_1b_2\bullet 1\}$$

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

$$N = N_1 + N_2.$$

$$\begin{aligned}
|\{b_1b_2b_3\}| &= |\{b_1b_2\bullet 0\}| + |\{b_1b_2\bullet 1\}| && \text{(sum rule)}\\
&= |\{b_1b_2\}| \times 2\\
&= (|\{b_1\bullet 0\}| + |\{b_1\bullet 1\}|) \times 2 && \text{(sum rule)}\\
&= |\{b_1\}| \times 2 \times 2\\
&= 2 \times 2 \times 2
\end{aligned}$$

Product Rule

$$|\{\, b_1 \quad b_2 \quad b_3 \,\}|$$
$$2 \times 2 \times 2$$

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_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* $x_1x_2x_3$;
⋮
Let N_r be the number of choices for x_r *after you choose* $x_1x_2x_3 \cdots x_{r-1}$.

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

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

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

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

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

4. **Passwords.** Use: {a,. . . ,z}, {A,. . . ,Z}, {0,. . . ,9}, special: {!,@,#,$,%,∧,&,*,(,)}.
   Rules: Length is 8. Must have at least one special.
   $$\begin{aligned}
   |\{\text{passwords}\}| &= 72 \times 72 \times \cdots \times 72 = 72^8 && \text{(product rule)}\\
   &= |\{\text{valid}\}| + |\{\text{invalid}\}| && \text{(sum rule)}\\
   &= |\{\text{valid}\}| + 62^8 && \text{(product rule)}
   \end{aligned}$$
   $|\{\text{valid}\}| = 72^8 - 62^8 \approx 5 \times 10^{14}$. (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: $2 \times 2 \times \cdots \times 2 = 2^{10} = 1024$.
   $|\{\text{committees}\}| = |\{\text{10-bit binary strings}\}|$

   e.g. $\{\, s_1\, s_2\, s_3\, s_4\, s_5\, s_6\, s_7\, s_8\, s_9\, s_{10} \,\}$
   1 1 0 1 0 0 0 0 1 0 ↔ $\{s_1, s_2, s_4, s_9\}$
   0 0 0 0 0 0 0 0 0 0 ↔ $\varnothing$

Examples from CS

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: ∼ 13TB)
  - 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 usinig 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.

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

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

$\binom{n}{k}$ (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	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

Pascal's Triangle

$$\{n\text{-sequence with } k \text{ 1's}\} = 0\bullet \underbrace{\{(n-1)\text{-sequence with } k \text{ 1's}\}}_{\binom{n-1}{k}} \;\cup\; 1\bullet \underbrace{\{(n-1)\text{-sequence with } (k-1) \text{ 1's}\}}_{\binom{n-1}{k-1}}$$

$$\boxed{\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}} \quad \text{(sum rule)} \qquad \text{base cases: } \binom{n}{0} = 1;\ \binom{n}{n} = 1.$$

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.

0🔴	1🔴	2🔴	3🔴	⋯	n🔴
$Q(n,k-1)$	$Q(n-1,k-1)$	$Q(n-2,k-1)$	$Q(n-3,k-1)$	⋯	$Q(0,k-1)$

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

$Q(n,k)$ (n \ k)	1	2	3	4	5	6	7	8	9	10	11
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

$Q(n,1) = 1;\ Q(0,k) = 1$

Challenge problems we had earlier.

1. (5 candies, 10 colors) → 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 $Q(5,10) = 2002$.

Counting One Set By Counting Another: Bijection

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

{🔴🔴🔴}	{🔴🔴🔵}	{🔴🔴🟢}	{🔴🔵🔵}	{🔴🔵🟢}	{🔴🟢🟢}	{🔵🔵🔵}	{🔵🔵🟢}	{🔵🟢🟢}	{🟢🟢🟢}
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
1	2	3	4	5	6	7	8	9	10

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

not a function | 1-to-1; **not** onto (injection, $A \overset{\text{INJ}}{\mapsto} B$) $|A| \le |B|$ | onto; **not** 1-to-1 (surjection, $A \overset{\text{SUR}}{\mapsto} B$) $|A| \ge |B|$ | 1-to-1 **and** onto (bijection, $A \overset{\text{BIJ}}{\mapsto} B$) $|A| = |B|$

$A \overset{\text{BIJ}}{\mapsto} B$ 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

3 colors, 🔴, 🔵, 🟢. Consider the 7-candy goody-bag {2🔴, 3🔵, 2🟢}:

red candies	delimiter	blue candies	delimiter	green candies		infer color from position
🔴🔴	|	🔵🔵🔵	|	🟢🟢	↔	○○ | ○○○ | ○○

goody-bags $\overset{\text{BIJ}}{\mapsto}$ 9-bit binary sequences with two 1's.

Examples.

00100010101000 → ○○|○○○|○|○|○○○ ↔ {2🔴, 3🔵, 1🟢, 1🟣, 3🟠}
1000011010000 → |○○○○||○|○○○○ ↔ {0🔴, 4🔵, 0🟢, 1🟣, 4🟠}

n candies and k colors → $(k-1)$ delimiters.

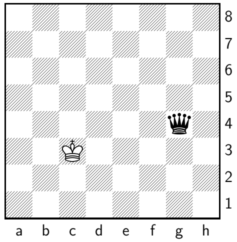
number of goody-bags with n candies of k colors = number of $(n+k-1)$-bit sequences with $(k-1)$ 1's

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

$\binom{n}{k}$ keeps popping up but we don't have a formula for it.

Nonaligned King-Queen Positions on a Chessboard

complex object	sequence to reconstruct it
king-queen position	$c3g4$
number of positions	number of sequences $c_K r_K c_Q r_Q$



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 $c_K \neq c_Q$.

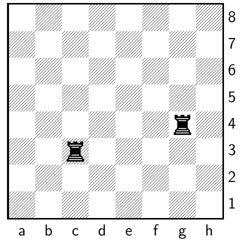
8 choices for c_K and r_K; after choosing c_K, r_K, there are only 7 choices for c_Q and r_Q.

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

complex object	sequence to reconstruct it
castle-castle position	$c3g4$
number of positions	number of sequences $c_1 r_1 c_2 r_2$



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 $|B| = k|A|$.

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)

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.

$$\begin{aligned} \text{number of } k\text{-orderings} &= \text{number of } k\text{-subsets} \times k! && \leftarrow \text{product rule} \\ &= \binom{n}{k} \times k! && \leftarrow \text{bijection to sequences with } k \text{ 1's} \end{aligned}$$

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

Exercise. How many 10-bit binary sequences with four 1's?

Binomial Theorem: $(x+y)^n = \sum_{i=1}^{n} \binom{n}{i} x^i y^{n-i}$

$$\begin{aligned} (x+y)^3 = (x+y)(x+y)(x+y) &= xxx + xxy + xyx + xyy + yxx + yxy + yyx + yyy \\ &= x^3 + 3x^2y + 3xy^2 + y^3. \end{aligned}$$

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

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

↑ strings with $(n-1)$ x's: $\binom{n}{n-1}$; ↑ strings with $(n-2)$ x's: $\binom{n}{n-2}$; ↑ strings with $(n-3)$ x's: $\binom{n}{n-3}$; ↑ strings with 1 x: $\binom{n}{1}$

$$= \binom{n}{n}x^n + \binom{n}{n-1}x^{n-1}y + \binom{n}{n-2}x^{n-2}y^2 + \binom{n}{n-3}x^{n-3}y^3 + \cdots + \binom{n}{1}xy^{n-1} + \binom{n}{0}y^n$$

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.