CSCI 2200
FOUNDATIONS OF COMPUTER SCIENCE

David Goldschmidt
goldsd3@rpi.edu
Fall 2022

COUNTING SEQUENCES – SUM & PRODUCT RULES

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

$$N = N_1 + N_2.$$

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

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

How many binary sequences of length n contain exactly k ones (with $0 \le k \le n$)…?

$\binom{n}{k}$ = number of n-length binary sequences with exactly k ones

Length 3:
000 001 010 011 100 101 110 111

Length 4: (the rightmost digits are the same in each line)
0000 0001 0010 0011 0100 0101 0110 0111
1000 1001 1010 1011 1100 1101 1110 1111

Length 5:
00000 00001 00010 00011 00100 00101 00110 …

$\binom{n}{k}$	k: 0	1	2	3	4	5	6	7	8
n: 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

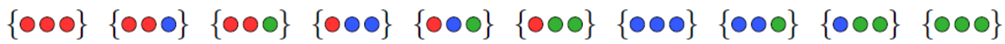
$$\{n\text{-sequence with } k \text{ 1's}\} = 0\bullet\underbrace{\{(n-1)\text{-sequence with } k \text{ 1's}\}}_{\binom{n-1}{k}} \;\cup\; 1\bullet\underbrace{\{(n-1)\text{-sequence with } (k-1) \text{ 1's}\}}_{\binom{n-1}{k-1}}$$

i.e., $\boxed{\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}}$ with base cases: $\binom{n}{0} = 1;\ \binom{n}{n} = 1.$

BUILD-UP COUNTING – EXAMPLE

Three different types of candy: (red, blue, green) (order does not matter…)

A goody-bag holds 3 candies — how many possible goody-bags are there?

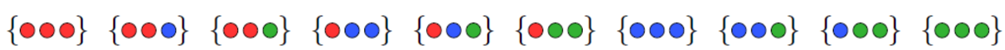


i.e., { AAA, AAB, AAC, ABB, ABC, ACC, BBB, BBC, BCC, CCC } — 10 possibilities

Generalize this: Q(n, k) = # of goody-bags of n candies with k colors — e.g., Q(3,3) = 10

Three different types of candy: (red, blue, green) (order does not matter…)

A goody-bag holds 3 candies — how many possible goody-bags are there?



i.e., { AAA, AAB, AAC, ABB, ABC, ACC, BBB, BBC, BCC, CCC } — 10 possibilities

Generalize this: Q(n, k) = # of goody-bags of n candies with k colors — e.g., Q(3,3) = 10

Start simple — if k = 1, then regardless of n, we have Q(n, 1) = 1…

…and if n = 0, then regardless of k, we have Q(0, k) = 1 (always one empty goody-bag)

$Q(n,k)$		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	1										
n	3	1										
	4	1										
	5	1										

Three different types of candy: (red, blue, green) (order does not matter…)

A goody-bag holds 3 candies — how many possible goody-bags are there?

i.e., { AAA, AAB, AAC, ABB, ABC, ACC, BBB, BBC, BCC, CCC } — 10 possibilities

Generalize this: Q(n, k) = # of goody-bags of n candies with k colors — e.g., Q(3,3) = 10

For build-up counting, assume i red candies ($i \geq 0$) — how do we pick the remaining candies?

$$Q(n, k) = i + Q(n - i, k - 1)$$

(loop through i red candies)

Let i = { 0, 1, …, n } — then there are (n + 1) types of goody-bags

we can use the sum rule…!

$$Q(n,k) = \sum_{i=0}^{n} Q(n-i, k-1) = \sum_{j=0}^{n} Q(j, k-1)$$

Three different types of candy: (red, blue, green) (order does not matter…)

A goody-bag holds 3 candies — how many possible goody-bags are there?

i.e., { AAA, AAB, AAC, ABB, ABC, ACC, BBB, BBC, BCC, CCC } — 10 possibilities

Generalize this: Q(n, k) = # of goody-bags of n candies with k colors — e.g., Q(3,3) = 10

For build-up counting, assume i red candies ($i \geq 0$) — how do we pick the remaining candies?

$$Q(n, k) = i + Q(n - i, k - 1)$$

(loop through i red candies)

Let i = { 0, 1, …, n } — then there are (n + 1) types of goody-bags

$$Q(n,k) = \sum_{i=0}^{n} Q(n-i, k-1) = \sum_{j=0}^{n} Q(j, k-1)$$

$Q(n,k)$		1	2
n	0	1	1
	1	1	2
	2	1	3
	3	1	4
	4	1	5
	5	1	6

Three different types of candy: ●●● (red, blue, green)

(order does not matter…)

A goody-bag holds 3 candies — how many possible goody-bags are there?

{●●●} {●●●} {●●●} {●●●} {●●●} {●●●} {●●●} {●●●} {●●●} {●●●}

i.e., { AAA, AAB, AAC, ABB, ABC, ACC, BBB, BBC, BCC, CCC } — 10 possibilities

Generalize this: Q(*n*, *k*) = # of goody-bags of *n* candies with *k* colors — e.g., Q(3,3) = 10

e.g., Q(3,3) = Q(0,2) + Q(1,2) + Q(2,2) + Q(3,2) = 1 + 2 + 3 + 4 = 10

$Q(n,k)$	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						1287	2002	3003

What does Q(5,10) mean…?

Generalize this: Q(*n*, *k*) = # of goody-bags of *n* candies with *k* colors — e.g., Q(3,3) = 10

e.g., Q(3,3) = Q(0,2) + Q(1,2) + Q(2,2) + Q(3,2) = 1 + 2 + 3 + 4 = 10

$Q(n,k)$	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

Can you prove that Q(*n*, *k*) = Q(*n*, *k* – 1) + Q(*n* – 1, *k*)…?

(See Exercise 13.6…)

These patterns should look familiar…

$\binom{n}{k}$	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	1	5	10	10	5	1			

Pascal's Triangle

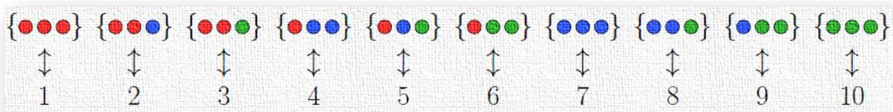
BIJECTION – (MAPPING PROBLEM A TO PROBLEM B)

Labeling goes a long way toward mapping one problem to another…

Let set A be the set of all goody-bags of 3 candies with 3 colors

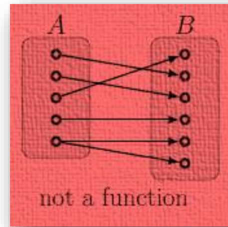
Let set $B = \{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$

The *bijection* of A to B maps each element of A to a distinct element of B:



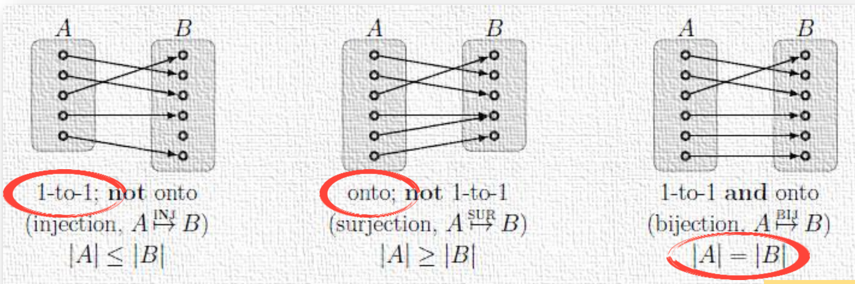
In a bijection, $|A| = |B|$ — and we can count A by counting B (or vice versa)

BIJECTION, INJECTION, AND SURJECTION



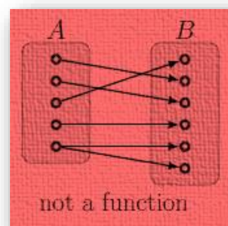
A function maps each element of set *A* to exactly one element of set *B*

We define *bijection*, *injection*, and *surjection* based on whether there is a *1-to-1 correspondence* or an *onto correspondence* — these are illustrated below:



See Pop Quiz 13.9...

BIJECTION, INJECTION, AND SURJECTION

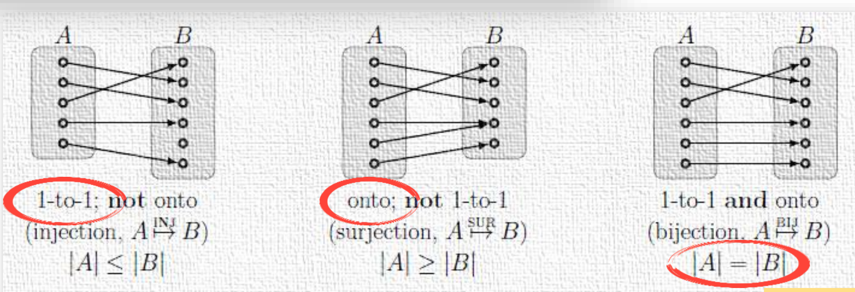


Clarifications:

- a 1-to-1 correspondence means every element in *A* maps to an element in *B*
- an onto correspondence means every element in *B* is mapped to by an element of *A*

A func... ...ent of set *B*

We de... ...ther there is a *1-to-1 correspondence or an onto correspondence* — these are illustrated below:



See Pop Quiz 13.9...

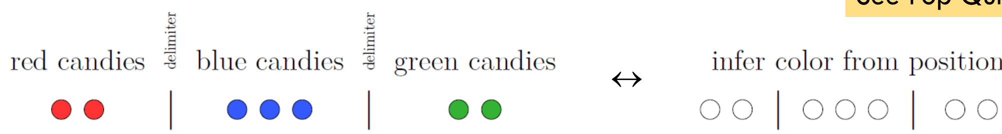
BIJECTION – (MAPPING PROBLEM A TO PROBLEM B)

Problem A: How many goody-bags of 7 candies with 3 possible colors are there…?

e.g., {2●, 3●, 2●} — consider the goody-bag containing 2 reds, 3 blues, and 2 greens…

…we can rewrite this and encode as follows:

See Pop Quiz 13.10…



Aha, this looks like binary sequence 001000100 — i.e., a 9-bit sequence with two 1-bits

Problem B: How many 9-bit sequences contain exactly two 1-bits…?

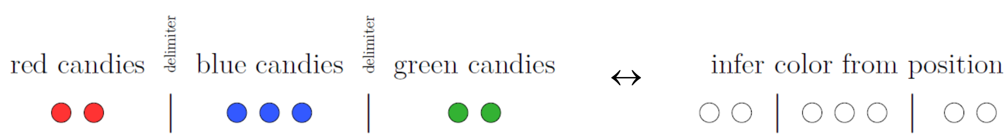
We mapped one problem to another through bijection: $A \xrightarrow{\text{BIJ}} B$

How can we generalize this bijection…?

BIJECTION – (MAPPING PROBLEM A TO PROBLEM B)

Problem A: How many goody-bags of 7 candies with 3 possible colors are there…?

e.g., {2●, 3●, 2●} — consider the goody-bag containing 2 reds, 3 blues, and 2 greens…



Problem B: How many 9-bit sequences contain exactly two 1-bits…?

How can we generalize this bijection…?

Examples with 5 colors and 10 or 9 candies, respectively…

00100010101000 → ○○|○○○|○|○|○○○ ↔ {2●, 3●, 1●, 1●, 3●}

1000011010000 → |○○○○||○|○○○○ ↔ {0●, 4●, 0●, 1●, 4●}

BIJECTION – (MAPPING PROBLEM A TO PROBLEM B)

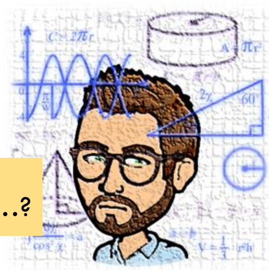
Problem A: How many goody-bags of n candies with k possible colors are there…?

Problem B: How many $(n + k - 1)$-bit sequences contain exactly $(k - 1)$ 1-bits…?

Solving either problem solves both problems, i.e., $A \xrightarrow{\text{BIJ}} B$

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

How can we define $\binom{n}{k}$…?



COMBINATIONS VS. PERMUTATIONS

Does order matter…?

We often want to select k objects from a set of n elements — sometimes, order matters

If order does not matter, we have a k-combination or k-subset

If order matters, we have a k-permutation or k-ordering

e.g., let $S = \{ a, b, c, d \}$ and let $k = 2$

How can we combine elements of S when order does not matter?

The 2-subsets are:

$\{ ab, ac, ad, bc, bd, cd \}$

How can we permutate elements of S when order does matter?

The 2-orderings are:

$\{ ab, ac, ad, ba, bc, bd, ca, cb, cd, da, db, dc \}$

k-ORDERINGS – PERMUTATIONS

How many k-orderings or k-permutations are there for a given set S...?

Given set S with $|S| = n$, we can use the product rule to count the k-orderings...

$$\#\text{ of } k\text{-orderings} = n \times (n-1) \times (n-2) \times \ldots \times (n-(k-1))$$

$$= \frac{n!}{(n-k)!}$$

With 10 runners in a race, how many possible top-3 finishes are there,
i.e., how many ways can runners come in first (F), second (S), and third (T)?

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

original solution using the product rule

$$\frac{10!}{(10-3)!} = \frac{10!}{7!} = 10 \times 9 \times 8 = 720$$

solution using k-orderings formula

k-SUBSETS – COMBINATIONS

How many k-subsets or k-combinations are there for a given set S...?

Given set S with $|S| = n$, we can first count the k-orderings using the product rule...

$$\#\text{ of } k\text{-orderings} = (\#\text{ of ways to pick } k\text{-subset } A) \times (\#\text{ of ways to order } A \text{ once picked})$$

A bijection exists between binary sequences with k 1-bits and k-subsets See Exercise 13.11(f)...

$\#$ of k-subsets $= \binom{n}{k}$ — and there are $k!$ ways to order k-subset A once picked

Thus, $\frac{n!}{(n-k)!} = \binom{n}{k} \times k!$ — i.e., $\boxed{\text{number of } k\text{-subsets} = \binom{n}{k} = \frac{n!}{k!(n-k)!}}$

How many 10-bit binary sequences have exactly four 1-bits?

COUNTING k-SUBSETS

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

How many 10-bit binary sequences have exactly four 1-bits?

This is a *k*-subset problem — from 10 bit positions, choose 4 bits to be 1…

Answer is $\binom{10}{4} = \frac{n!}{k!(n-k)!} = \frac{10!}{4!(10-4)!} = \frac{10 \times 9 \times 8 \times 7}{4 \times 3 \times 2} = 210$

How many 10-bit binary sequences have at most four 1-bits?

How many 10-bit binary sequences have at least four 1-bits?

How many 10-bit binary sequences have an equal number of 0- and 1-bits?

break these down into subproblems, then sum the counts for each one…

…also look for symmetry and shortcuts

Notice the relationship (factor of *k*!) between *k*-subsets and *k*-orderings…

```
Given S = { a, b, c, d }
n = 4
k = 2

2-subsets: { ab, ac, ad, bc, bd, cd }

  4          4!            4 x 3
(   )  =  ----------  =  ---------  =  6
  2        2!(4-2)!          2

2-orderings:
 { ab, ac, ad, ba, bc, bd, ca, cb, cd, da, db, dc }

            4!           4 x 3
         --------  =  ---------  =  12
          (4-2)!           1
```

Tinker with these smaller examples by slowly increasing *k* and *n*…

COUNTING CARDS — POKER HANDS

A standard deck of playing cards has 52 cards

Each card has a value { A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K } and a suit {♠, ♥, ♦, ♣}

A poker hand is 5 cards, e.g., 5♣, 8♣, 10♦, 5♦, J♥ — order does not matter

Count: (a) All poker hands — $\binom{52}{5} = \frac{52!}{5!(52-5)!} = \frac{52 \times 51 \times 50 \times 49 \times 48}{5 \times 4 \times 3 \times 2} = 2{,}598{,}960$

(b) 4-of-a-kind — how do we construct a hand? Pick a value for 4 cards… …then pick the 5th card — product rule gives us $13 \times 48 = 624$

(c) Flush (same suit)

(d) Full house (3-of-a-kind and a pair)

(e) Straight (e.g., 7, 8, 9, 10, J)

Complete these counting problems and see Exercise 13.14…

WHAT NEXT…?

Look at the schedule and plan your last few weeks…!

Exam 2 grading continues…
- …aim to have grades posted early next week

Probability lecture recordings and Problem Set 8 will be posted early next week…
- …with Problem Set 8 due in your recitations on Wednesday, November 30

After Thanksgiving, we will covers Models of Computation, i.e., Chapters 23-29…
- …with Homework 5 due 11:59PM Thursday, December 8

* * * **Final Exam is scheduled 3:00-6:00PM on Wednesday, December 14** * * *

Practice! Tinker! Practice! Tinker! Practice! Tinker! Practice! Tinker! Practice!