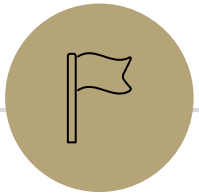
Even *More* Counting

CSE 312 26Su

Lecture 3

Fun Fact

The number of possible unique chess games is estimated to be around 10^{120}, which is more than the number of atoms in the universe! To get an idea of how this number gets so big – the white pieces have 20 options for the first move, and the black pieces has 20 options for the second move. After only this first play, there are $20 \cdot 20 = 400$ possibilities. Imagine how this increases after more moves!



Quiz 1

Announcements

> HW1 is out and due on Wednesday

> Remember to do concept checks!

> Office hours starts today, after lecture

> Also, please start on the homework early. Remember you need to have a meaningful attempt to utilize the penalty free late days.

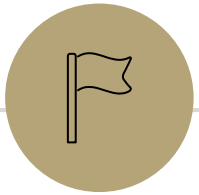
Where Are We?

Last time:

- More about combinations
- Applications of combinations
  - Path counting
  - Rearranging strings
    Multinomial coefficients
  - Binomial theorem
- Inclusion-exclusion
- Sleuth's criterion

Today:

- Stars and Bars
- Combinatorial Proofs
- Pigeonhole Principle



Stars and Bars

The last counting rule in a while! :')

Cupcakes

You’re buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

Cupcakes

You’re buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There’s vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

Cupcakes

You’re buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There’s vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

> There are unlimited cupcakes of each type
> Cupcakes of the same type are indistinguishable
> Order of the cupcakes in the box does not matter

Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

> There are unlimited cupcakes of each type
> Cupcakes of the same type are indistinguishable
> Order of the cupcakes in the box does not matter

Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

> There are unlimited cupcakes of each type
> Cupcakes of the same type are indistinguishable
> Order of the cupcakes in the box does not matter

Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

Idea: 5^{12} -- choose one of 5 types for each of the 12 cupcakes

Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

Idea: 5^{12} -- choose one of 5 types for each of the 12 cupcakes

Cupcakes (an incorrect approach)

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

Idea: 5^{12} -- choose one of 5 types for each of the 12 cupcakes

But the order of the cupcakes in the box doesn't matter so these should be considered the same outcome.

Cupcakes (an incorrect approach)

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

Idea: 5^{12} -- choose one of 5 types for each of the 12 cupcakes

Q: is div by 12! enough?

2 vanilla, 1 PB, 3 mint, 2 confetti, 4 rainbow

Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

All we care about is how many of each cupcake is ordered!

2 vanilla, 1 PB, 3 mint, 2 confetti, 4 rainbow

Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

All we care about is how many of each cupcake is ordered!

2 vanilla, 1 PB, 3 mint, 2 confetti, 4 rainbow

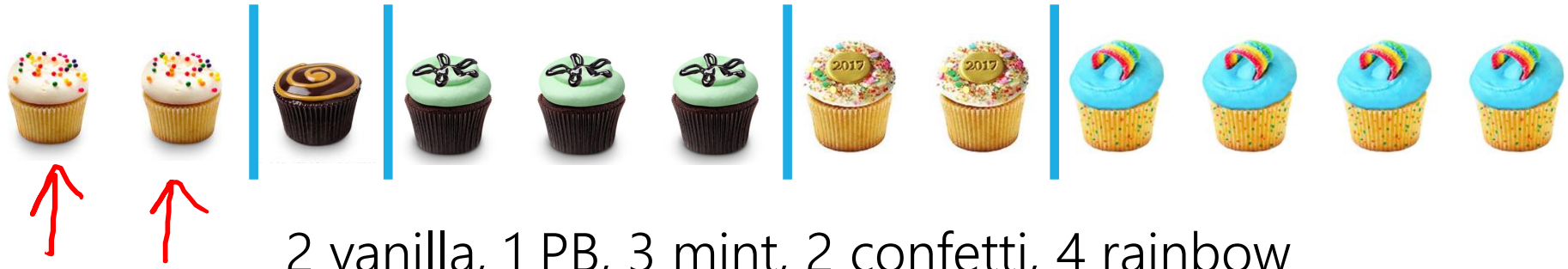
Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

All we care about is how many of each cupcake is ordered!



2 vanilla, 1 PB, 3 mint, 2 confetti, 4 rainbow

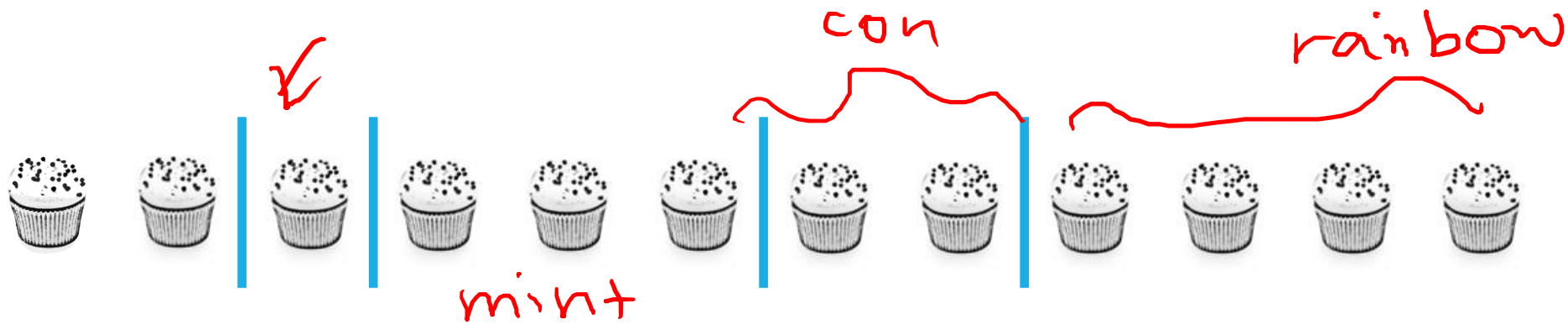
Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

All we care about is how many of each cupcake is ordered!



2 vanilla, 1 PB, 3 mint, 2 confetti, 4 rainbow

Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

We only need $\mathbf{5-1=4}$ **dividers to divide** this set of 12 cupcakes **into 5 groups**

2 vanilla, 1 PB, 3 mint, 2 confetti, 4 rainbow

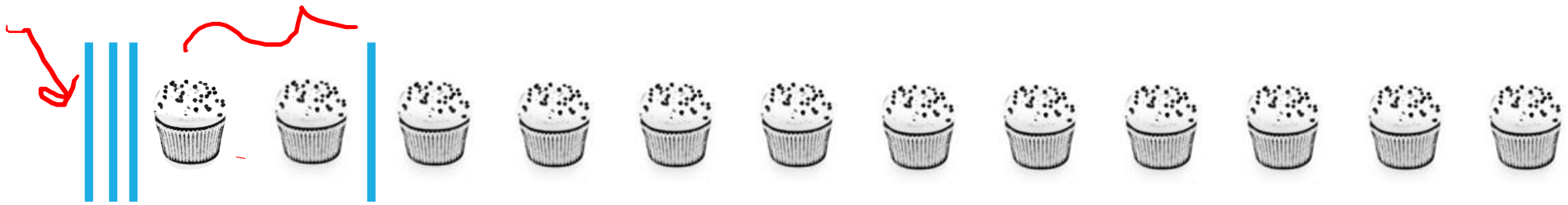
Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

We only need $\mathbf{5-1=4}$ **dividers to divide** this set of 12 cupcakes **into 5 groups**



0 vanilla, 0 PB, 0 mint, 2 confetti, 10 rainbow

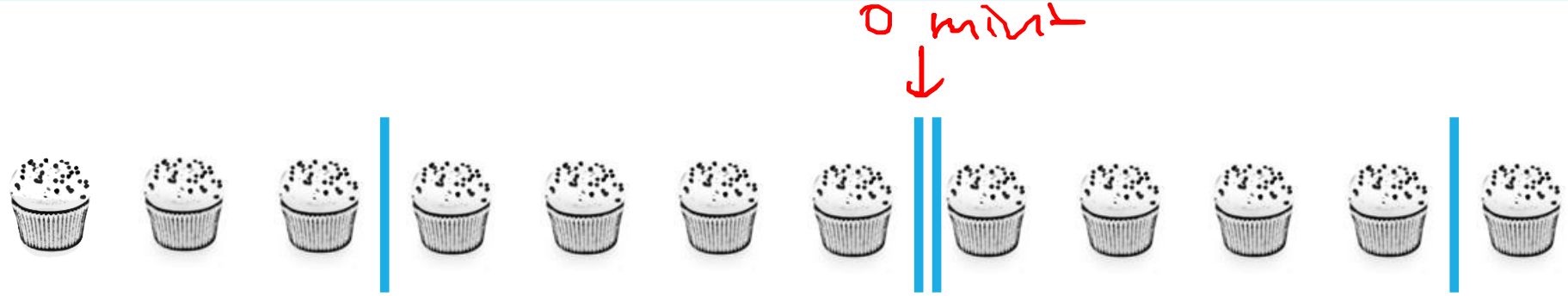
Cupcakes

You're buying one-dozen cupcakes (i.e., 12 cupcakes) for a party.

There's vanilla, peanut butter, mint, confetti, and rainbow (i.e., 5 types)

How many different cupcake boxes can you buy?

We only need $\mathbf{5-1=4}$ **dividers to divide** this set of 12 cupcakes **into 5 groups**



3 vanilla, 4 PB, 0 mint, 4 confetti, 1 rainbow

Cupcakes

You're buying cupcakes (i.e., 12 cupcakes) from vanilla, PB, mint, confetti, and rainbow (i.e., 5 types). How many different boxes can you buy?

1. Define an arbitrary ordering of the flavors
   in our example, our ordering is vanilla, PB, mint, confetti, rainbow
2. Place bars between the cupcakes to demarcate flavor

3 vanilla, 4 PB, 0 mint, 4 confetti, 1 rainbow

Cupcakes

You're buying cupcakes (i.e., 12 cupcakes) from vanilla, PB, mint, confetti, and rainbow (i.e., 5 types). How many different boxes can you buy?

1. Define an arbitrary ordering of the flavors
   in our example, our ordering is vanilla, PB, mint, confetti, rainbow
2. Place bars between the cupcakes to demarcate flavor

= How many ways to assign 4 dividers among the 12 cupcakes?

3 vanilla, 4 PB, 0 mint, 4 confetti, 1 rainbow

Cupcakes = A String Rearrangement

How many ways to assign 4 dividers among the 12 cupcakes?

Approach 1:

This is a string rearranging problem!

String with 12 identical C's (cupcakes) and 4 identical |'s (dividers)

e.g., "CCC|CCCC||CCCC|C", "CCCCCCCCCCCC||||"

1. Arrange letters in the string: $(12 + 4)! = 16!$

2. Divide out overcounting for duplicate letters: $\frac{16!}{12!4!} = \binom{16}{12}$

Cupcakes = Placing Dividers

How many ways to assign 4 dividers among the 12 cupcakes?

Approach 2:

There are $12 + 4 = 16$ positions in total. 4 of these will be a divider and the remaining 12 will be cupcakes

Cupcakes = Placing Dividers

How many ways to assign 4 dividers among the 12 cupcakes?

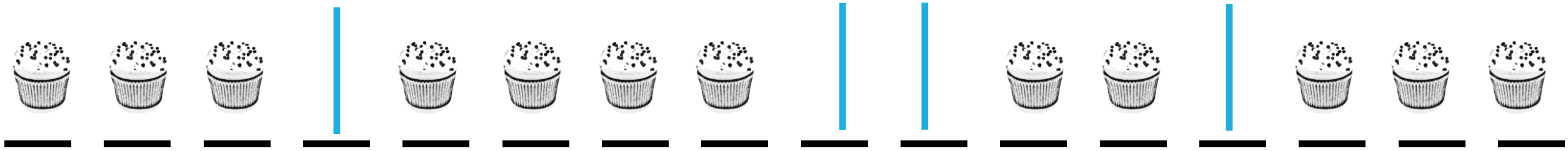
Approach 2:

There are $12 + 4 = 16$ positions in total. 4 of these will be a divider and the remaining 12 will be cupcakes

1. Pick 12 of the 16 positions to be cupcakes: $\binom{16}{12}$

Cupcakes = Placing Dividers

How many ways to assign 4 dividers among the 12 cupcakes?



Approach 2:

There are $12 + 4 = 16$ positions in total. 4 of these will be a divider and the remaining 12 will be cupcakes

1. Pick 12 of the 16 positions to be cupcakes: $\binom{16}{12}$
2. Pick 4 of the remaining 4 positions to be dividers: $\binom{4}{4} = 1$

Cupcakes = Placing Dividers

How many ways to assign 4 dividers among the 12 cupcakes?

Approach 2:

There are $12 + 4 = 16$ positions in total. 4 of these will be a divider and the remaining 12 will be cupcakes

1. Pick 12 of the 16 positions to be cupcakes: $\binom{16}{12}$
2. Pick 4 of the remaining 4 positions to be dividers: $\binom{4}{4} = 1$

$\binom{16}{12}$ ways in total – same as with previous approach!

In General

k-1 bars

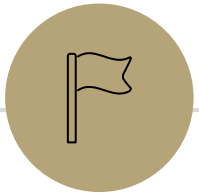
Stars and Bars

To pick n objects from k groups (where **order doesn't matter** and every **element of each group is indistinguishable**), use the formula:

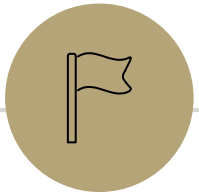
$$\binom{n+k-1}{n} = \binom{n+k-1}{k-1}$$

This counting technique is often called "stars and bars"
using a "star" instead of a cupcake, and calling the dividers "bars"

> What often hints to use the stars-and-bars approach is when we're dealing with picking a set from objects where objects from the same types are **indistinguishable**



Proofs Using Counting Techniques



Combinatorial Proofs

Let's first look at some combination facts...

Reminder - k-combination

k-combination

The number of k-element subsets from a set of n symbols is:

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

Some Facts about combinations

Here are some known facts about combinations:

- **Symmetry of combinations**: $\binom{n}{k} = \binom{n}{n-k}$
- **Pascal's Rule**: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

Some Facts about combinations

Here are some known facts about combinations:

> Symmetry of combinations: $\binom{n}{k} = \binom{n}{n-k}$

> Pascal's Rule: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

How do we prove these equations are true?

First Proof of Symmetry ($\binom{n}{k} = \binom{n}{n-k}$)

Proof 1: By algebra

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$ Definition of Combination

$$= \frac{n!}{(n-k)!k!}$$ Algebra (commutativity of multiplication)

$$= \binom{n}{n-k}$$ Definition of Combination

First Proof of Symmetry ($\binom{n}{k} = \binom{n}{n-k}$)

Wasn't that a great proof.

Airtight. No disputing it.

Got to say "commutativity of multiplication."

But…do you know *why*? Can you *feel* why it's true?

Second Proof of Symmetry ($\binom{n}{k} = \binom{n}{n-k}$)

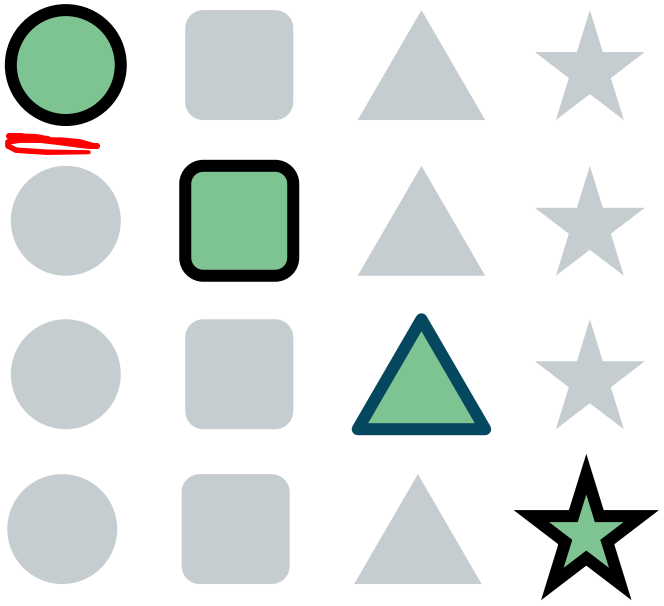
Both sides of this count the same set of outcomes! For example….

LHS: choose k things to include in the set

$n = 4$

$\binom{4}{1} \rightarrow$

$k = 1$



Second Proof of Symmetry ($\binom{n}{k} = \binom{n}{n-k}$)

Both sides of this count the same set of outcomes! For example….

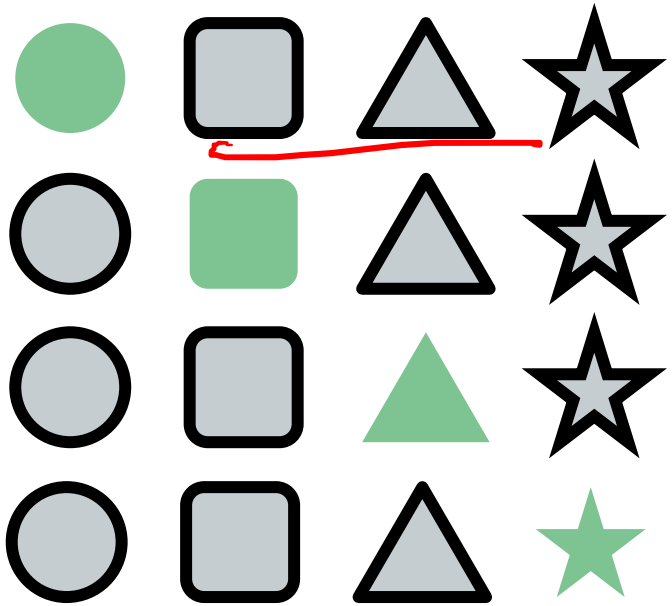
LHS: choose k things to include in the set

$\binom{4}{1} \rightarrow$

RHS: choose $n - k$ things to not include in the set

$n = 4$

$\binom{4}{3} \rightarrow$



Second Proof of Symmetry ($\binom{n}{k} = \binom{n}{n-k}$)

more formal – <u>combinatorial proof</u>

Suppose you have n people, and need to choose k people to be on your team. We will <u>count the number of possible teams two different ways</u>.

Way 1 (LHS): We choose the k people to be on the team. Since order doesn't matter (you're on the team or not), there are $\binom{n}{k}$ possible teams.

Second Proof of Symmetry ($\binom{n}{k} = \binom{n}{n-k}$)

more formal – <u>combinatorial proof</u>

Suppose you have n people, and need to choose k people to be on your team. We will <u>count the number of possible teams two different ways</u>.

Way 1 (LHS): We choose the k people to be on the team. Since order doesn't matter (you're on the team or not), there are $\binom{n}{k}$ possible teams.

Way 2 (RHS): We choose the $n-k$ people to <u>NOT</u> be on the team. Everyone else is on it. Since ~~order~~ again doesn't matter, there are $\binom{n}{n-k}$ possible ways to choose the team.

Second Proof of Symmetry ($\binom{n}{k} = \binom{n}{n-k}$)

more formal – <u>combinatorial proof</u>

Suppose you have n people, and need to choose k people to be on your team. We will <u>count the number of possible teams two different ways</u>.

Way 1 (LHS): We choose the k people to be on the team. Since order doesn't matter (you're on the team or not), there are $\binom{n}{k}$ possible teams.

Way 2 (RHS): We choose the $n-k$ people to NOT be on the team. Everyone else is on it. Since order again doesn't matter, there are $\binom{n}{n-k}$ possible ways to choose the team.

Since we're counting the same thing, the numbers must be equal.
So $\binom{n}{k} = \binom{n}{n-k}$.

Pascal's Rule: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

algebraic proof

$$\binom{n-1}{k-1} + \binom{n-1}{k} = \frac{(n-1)!}{(k-1)!(n-1-[k-1])!} + \frac{(n-1)!}{k!(n-1-k)!} \qquad \text{definition of combination}$$

$$= \frac{(n-1)!}{(k-1)!(n-k)!} + \frac{(n-1)!}{k!(n-k-1)!} \qquad \text{subtraction}$$

$$= \frac{[(n-1)!k!(n-k-1)!] + [(n-1)!(k-1)!(n-k)!]}{k!(k-1)!(n-k)!(n-k-1)!} \qquad \text{Find a common denominator}$$

$$= \frac{(n-1)!(k-1)!(n-k-1)!\,[k+(n-k)]}{k!(k-1)!(n-k)!(n-k-1)!} \qquad \text{factor out common terms}$$

$$= \frac{(n-1)!\,[k+(n-k)]}{k!(n-k)!} \qquad \text{Cancel } (k-1)!(n-k-1)!$$

$$= \frac{(n-1)!\cdot n}{k!(n-k)!} = \frac{n!}{k!(n-k)!} \qquad \text{Algebra}$$

$$= \binom{n}{k} \qquad \text{Definition of combination}$$

Pascal's Rule: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

combinatorial proof

Suppose you have n people (one of whom is Krisha), and need to choose k people to be on your team.

Pascal's Rule: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

combinatorial proof

Suppose you have n people (one of whom is Krisha), and need to choose k people to be on your team.

Way 1: There are n people total, of which we're choosing k (and since it's a team order doesn't matter) $\binom{n}{k}$.

Pascal's Rule: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

combinatorial proof

Suppose you have n people (one of whom is Krisha), and need to choose k people to be on your team.

Way 1: There are n people total, of which we're choosing k (and since it's a team order doesn't matter) $\binom{n}{k}$.

Way 2: There are two types of teams. Those for Krisha makes the team, and those for Krisha does not.

Pascal's Rule: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

combinatorial proof

Suppose you have n people (one of whom is Krisha), and need to choose k people to be on your team.

Way 1: There are n people total, of which we're choosing k (and since it's a team order doesn't matter) $\binom{n}{k}$.

Way 2: There are two types of teams. Those for Krisha makes the team, and those for Krisha does not.

Krisha + k-1 people

If Krisha does make the team, then $k-1$ of the other $n-1$ also make it.

If Krisha doesn't make the team, k of the other $n-1$ are on the team.

Pascal's Rule: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

combinatorial proof

Suppose you have n people (one of whom is Krisha), and need to choose k people to be on your team.

Way 1: There are n people total, of which we're choosing k (and since it's a team order doesn't matter) $\binom{n}{k}$.

Way 2: There are two types of teams. Those for Krisha makes the team, and those for Krisha does not.

If Krisha does make the team, then $k-1$ of the other $n-1$ also make it.

If Krisha doesn't make the team, k of the other $n-1$ are on the team.

Overall, by sum rule, $\binom{n-1}{k-1} + \binom{n-1}{k}$.

$\binom{n-2}{k-2} + \binom{n-2}{k} + \binom{n-2}{k-1}$

a b

Since we're computing the same number two different ways, they must be equal. So: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

Combinatorial Proofs

1. Describe a scenario
2. Explain how the LHS counts the outcomes in that scenario
3. Explain how the RHS counts the outcomes in that same scenario
4. State that "because the LHS and RHS both count the number of outcomes in the same scenario, they must be equal"

Combinatorial Proofs

When to use?

> Algebraic proofs are often difficult and don't give us an "intuitive" reason for why the equation holds

> ***Combinatorial proofs*** are used for proving an equation that may involve combinations, factorials, permutations, etc.

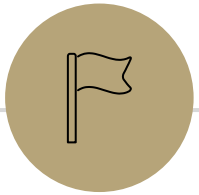
1. Describe a scenario
2. Explain how the LHS counts the outcomes in that scenario
3. Explain how the RHS counts the outcomes in that same scenario
4. State that "because the LHS and RHS both count the number of outcomes in the same scenario, they must be equal"

Combinatorial Proofs – tips!

> Start with the side that "looks simpler"

> Multiplying terms together indicates some sequential process (the product rule)

> A summation/adding terms indicates some disjoint cases that are added together using the sum rule



Pigeonhole Principle

A counting property that will be helpful in many proofs!

A Hairy Question

Are there at least two people in Seattle with the same number of hairs on their head?

A. Yes

B. No

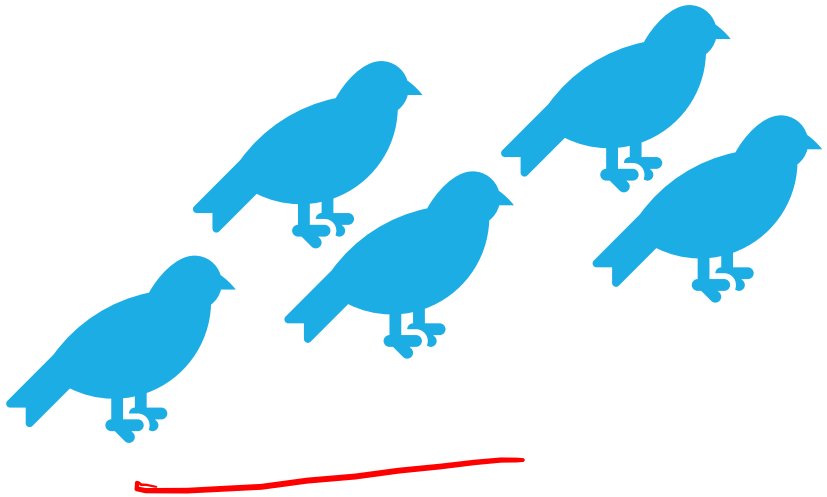
C. ???

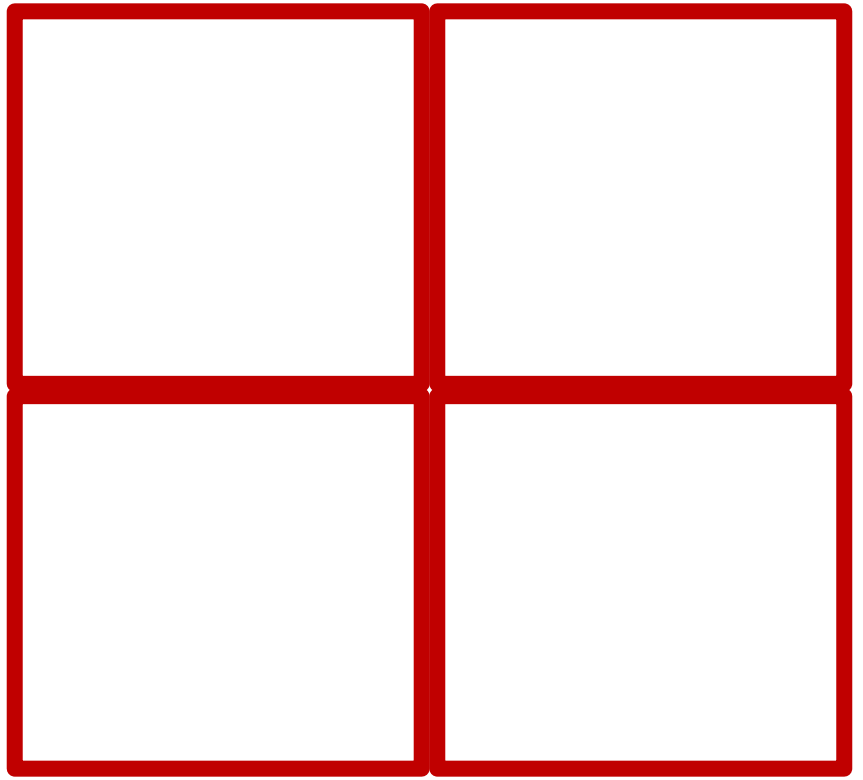
D. How would I know?

Join at **slido.com**
#3482 919

Pigeonhole Principle

If you have 5 pigeons, and place them into 4 holes, then…

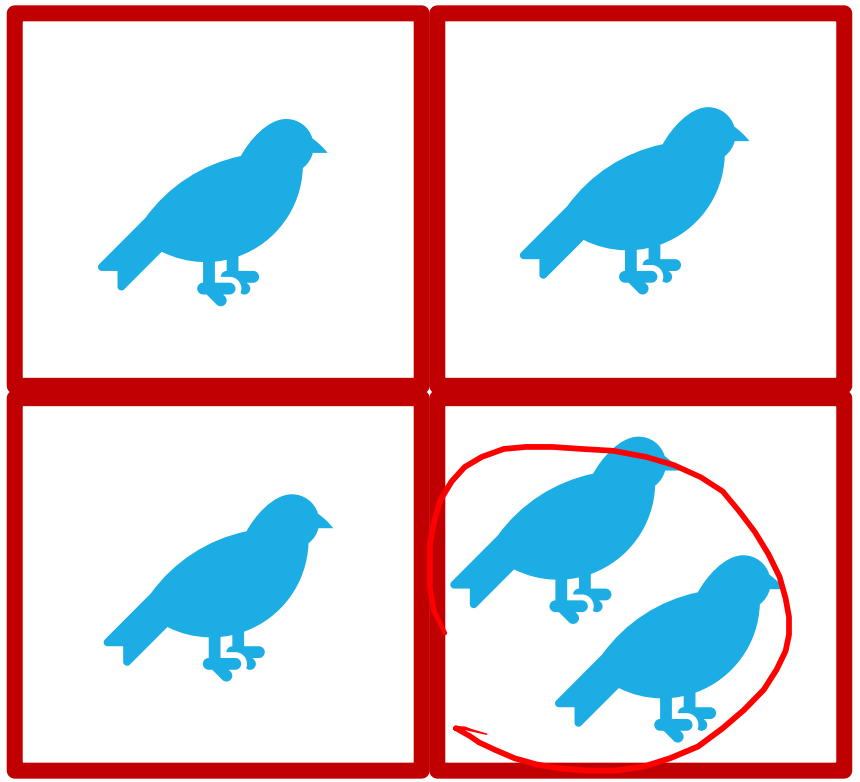




Pigeonhole Principle

If you have 5 pigeons, and place them into 4 holes, then…

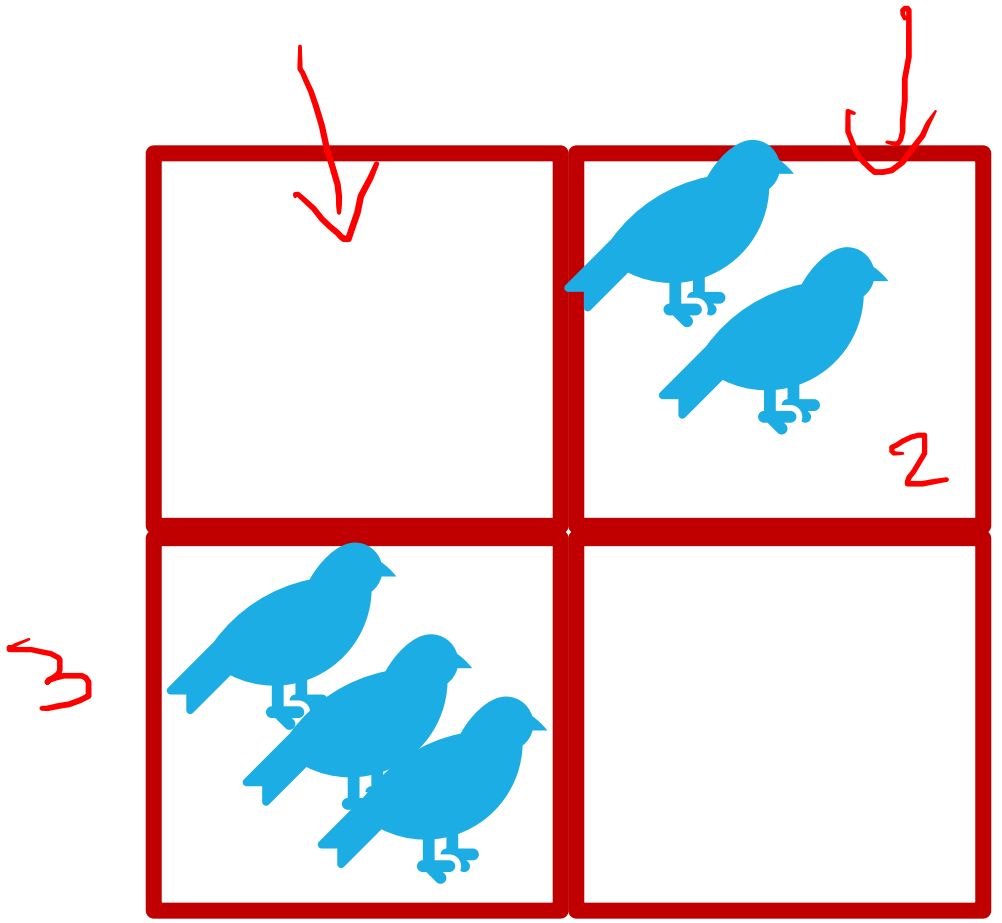
At least 2 pigeons are in the same hole.



Pigeonhole Principle

If you have 5 pigeons, and place them into 4 holes, then…

At least 2 pigeons are in the same hole.



Pigeonhole Principle

Pigeonhole Principle

Version 1
If you have n **pigeons**, and place them into k **holes**, and $n > k$, then **at least** 2 pigeons are in the same hole.

Note: we're not making guarantees about every pigeonhole – we're just guaranteeing that there is at least one hole with at least a certain number of pigeons

Pigeonhole Principle

Pigeonhole Principle

Version 1

If you have n **pigeons**, and place them into k **holes**, and $n > k$, then **at least** 2 pigeons are in the same hole.

Version 2 (generalized)

If you have n **pigeons**, and place them into k **holes**, and $n > k$, then **at least** $\left\lceil \frac{n}{k} \right\rceil$ pigeons are in the same hole.

10 3

$\left\lceil \frac{10}{3} \right\rceil = \lceil 3.337 \rceil = 4$

Note: we're not making guarantees about every pigeonhole – we're just guaranteeing that there is at least one hole with at least a certain number of pigeons

exists

Pigeonhole Principle (generalized)

If you have n pigeons and k pigeonholes, then there is at least one pigeonhole that has at least $\left\lceil \frac{n}{k} \right\rceil$ pigeons.

$\lceil a \rceil$ is the "ceiling" of a (it means always round up, $\lceil 1.1 \rceil = 2$, $\lceil 1 \rceil = 1$).

e.g., 5 pigeons and 4 holes -> at least 1 hole has $\left\lceil \frac{5}{4} \right\rceil = \lceil 1.25 \rceil = 2$ *pigeons*

An example

If you have to take 10 classes, and have 3 quarters to take them in, then…

An example

If you have to take 10 classes, and have 3 quarters to take them in, then…

1. **Pigeons**: The classes to take
2. **Pigeonholes**: The quarter
3. **Mapping**: Which class you take the quarter in
   (each class will be assigned to exactly 1 quarter)

An example

If you have to take 10 classes, and have 3 quarters to take them in, then…

Pigeons: The classes to take

Pigeonholes: The quarter

Mapping: Which class you take the quarter in
(each class will be assigned to exactly 1 quarter)

Applying the pigeonhole principle, there is at least one quarter where you take at least $\left\lceil \frac{10}{3} \right\rceil = 4$ courses.

An example

If you have to take 10 classes, and have 3 quarters to take them in, then…

Pigeons: The classes to take

Pigeonholes: The quarter

Mapping: Which class you take the quarter in
(each class will be assigned to exactly 1 quarter)

Remember!
When defining pigeons and pigeonholes, every pigeon **must** be assigned to one pigeonhole

Applying the pigeonhole principle, there is at least one quarter where you take at least $\left\lceil \frac{10}{3} \right\rceil = 4$ courses.

Practical Tips

1. What are the **pigeons?**
2. What are the **pigeonholes?**
3. How do you **map from pigeons to pigeonholes?**
4. There are n pigeons and k pigeonholes, so **by pigeonhole principle**, there is at least 1 pigeonhole with at least $\left\lceil \frac{n}{k} \right\rceil$ pigeons. This is what we're trying to prove because…

Practical Tips

When to use pigeonhole principle?

> When we're trying to **prove** or **make some guarantee** about a certain number of things sharing some property.

> For tricky ones, we'll warn you in advance that pigeonhole principle is the right method (you'll see one in the section handout).

1. What are the **pigeons?**
2. What are the **pigeonholes?**
3. How do you **map from pigeons to pigeonholes?**
4. There are n pigeons and k pigeonholes, so **by pigeonhole principle**, there is at least 1 pigeonhole with at least $\left\lceil \frac{n}{k} \right\rceil$ pigeons. This is what we're trying to prove because…

Practical Tips

Look for – a set you're trying to divide into groups, where collisions would help you somehow.

"Prove that there are at least **x A's** that have the same **B"**

> Pigeons: all possible A's

> Pigeonholes: all possible B's

> Mapping: each A is assigned to one possible B

The pigeonhole principle tells us that there are at least x pigeons that fall into the same hole -> there are at least x A's that have the same B

Back to the hairy question….

Are there at least two people in Seattle with the same number of hairs on their head?

Pigeons: All people in Seattle
780,000 people in Seattle (**780,000 pigeons**)

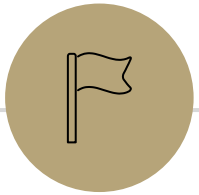
Pigeonholes: All possible 'hair counts' -> ~
people have up to 150,000 hairs on their head (**150,000 pigeonholes**)

Mapping: Every person is "assigned" to a particular hair count

By the **pigeonhole principle**, there are at least $\left\lceil \frac{780{,}000}{150{,}000} \right\rceil = 5$ pigeons in the same pigeonhole -> at least 5 people in Seattle with the same hair count!

But actually…what's the point of this?

- We can prove some pretty fun statements!
- **Hashing Algorithms**: Ensuring that collisions will occur when hashing more items than there are available hash slots, which helps in designing effective collision resolution strategies. *(you'll learn about hashing in 332!)*
- **Resource Allocation**: Ensuring that in systems with more requests than resources, some resources will be shared (e.g., tasks and processors)
- **Data Compression**: Demonstrating that lossless compression of sufficiently large files must lead to some files being larger after compression due to limited encoding space
- It is also the basis of **proving that a language is not regular**, as you may remember from 311!



Thinking about Counting

We've seen lots of ways to count

Sum rule (split into disjoint sets)

Product rule (use a sequential process)

Combinations (order doesn't matter)

Permutations (order does matter)

Principle of Inclusion-Exclusion (counting the size of a union of sets)

Complementary Counting (counting the ways for something to **not** occur)

"Stars and Bars" $\binom{n+k-1}{k-1}$ (assigning indistinguishable stars to distinguishable bins)

Niche Rules (useful in very specific circumstances)

Binomial Theorem

Pigeonhole Principle

How to tell which approach(es) to take?

- Identify keywords in the problem, and key properties, and think about what techniques match those patterns
- If an approach isn't working or things are getting out of control, try a different approach!
- There are often multiple ways to solve the same problem ☺
- In some harder problems, we might need to start by…
  - solving a simplified version of the problem first,
  - and then dividing/subtracting out overcounting

This all takes practice so don't be hard on yourself if you don't think of the correct answer at first glance!

How to tell which approach(es) to take?

> Can we break this into some disjoint cases? (**sum rule**)

> Can we create our desired outcomes with a clear sequential process? (**product rule**)

> Are we counting the ways for something not to occur? (**complementary counting**)

> Are we dealing with the union of some sets? (**inclusion-exclusion**)

> Does order matter or not in this problem? (**permutation** or **combination**)

> Are the objects distinguishable or indistinguishable (**stars and bars** if indistinguishable)

These approaches often work together! E.g., to compute the number of options in one of the disjoint cases, you may need another approach