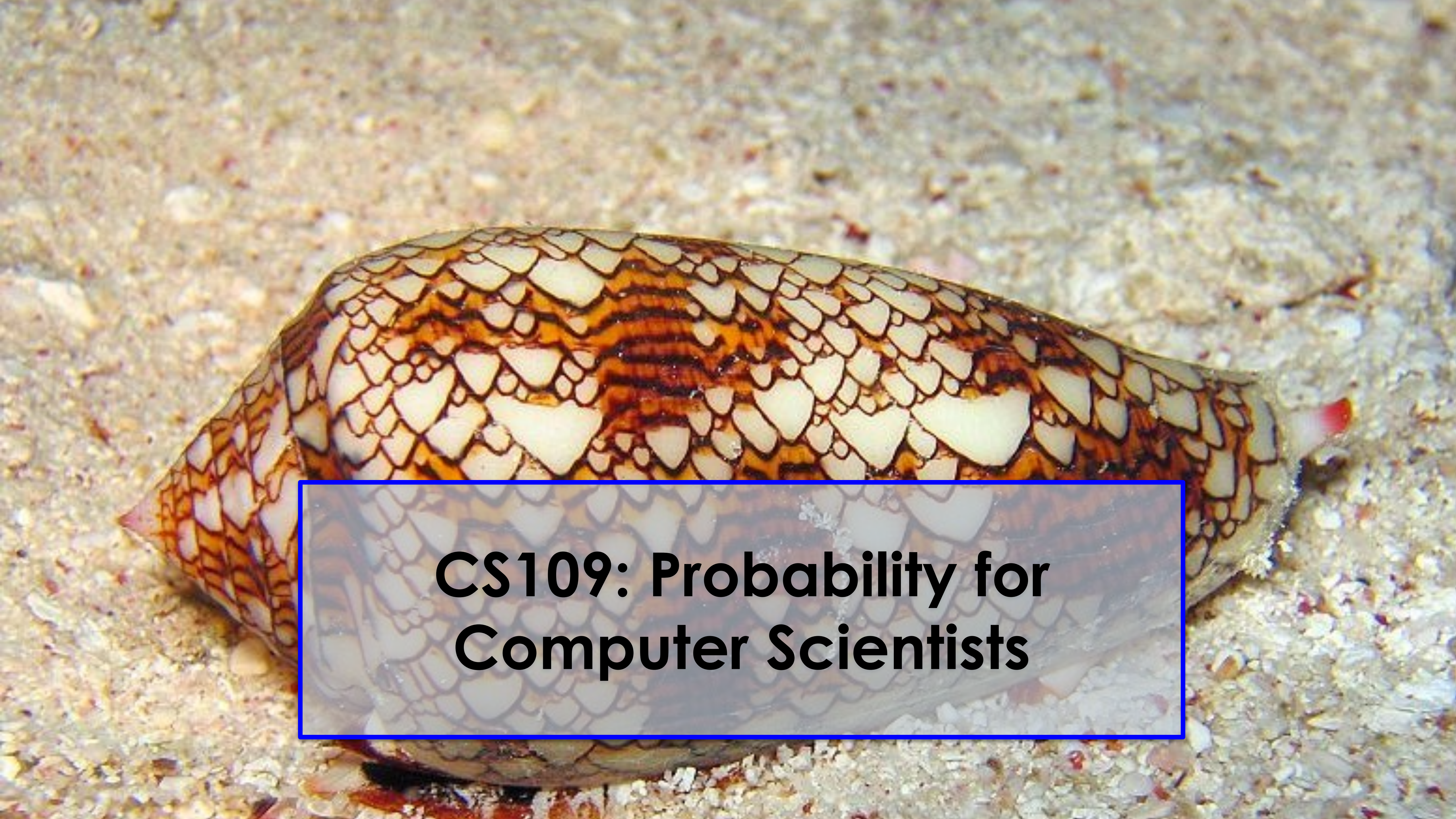
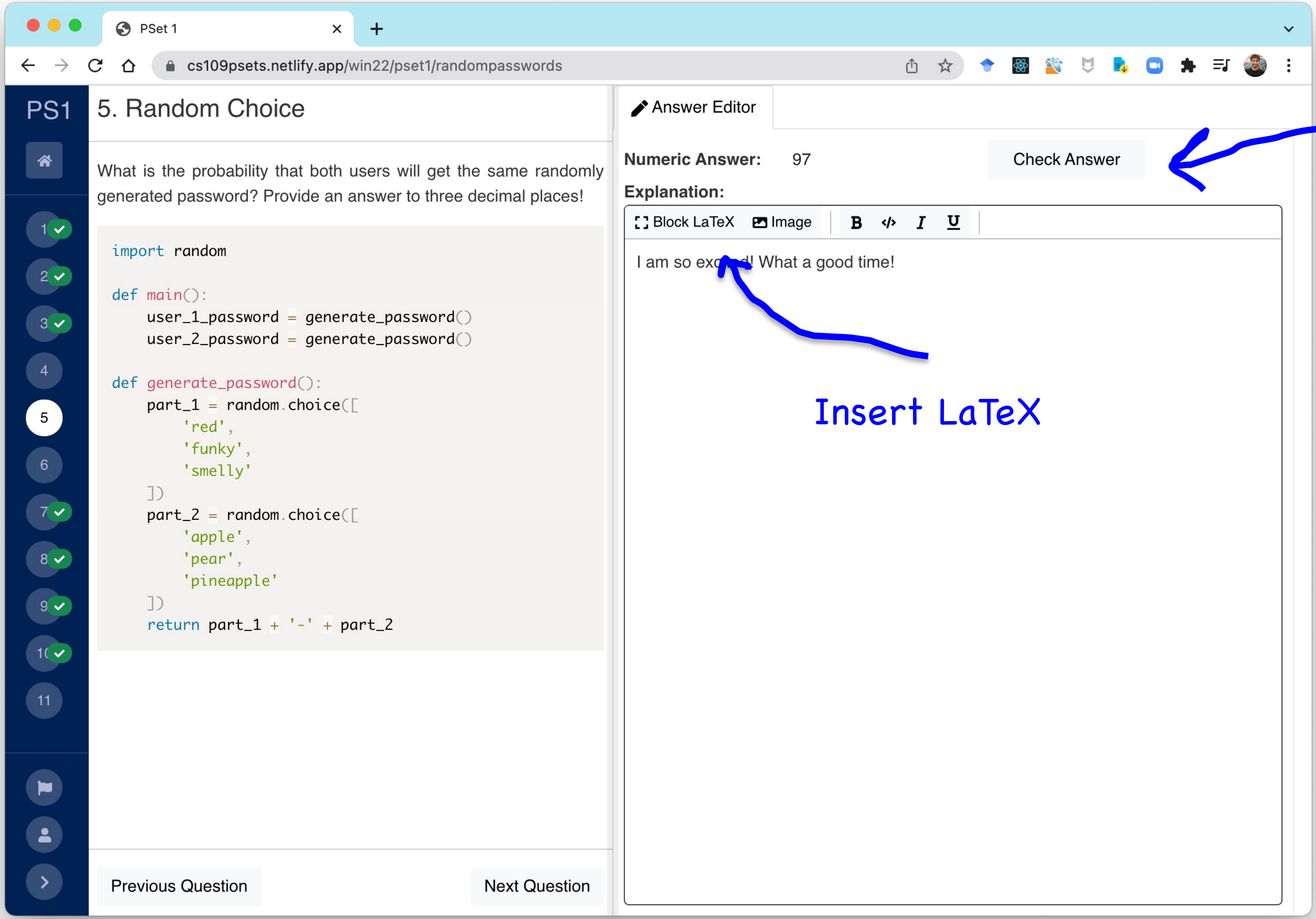


CS109: Probability for Computer Scientists

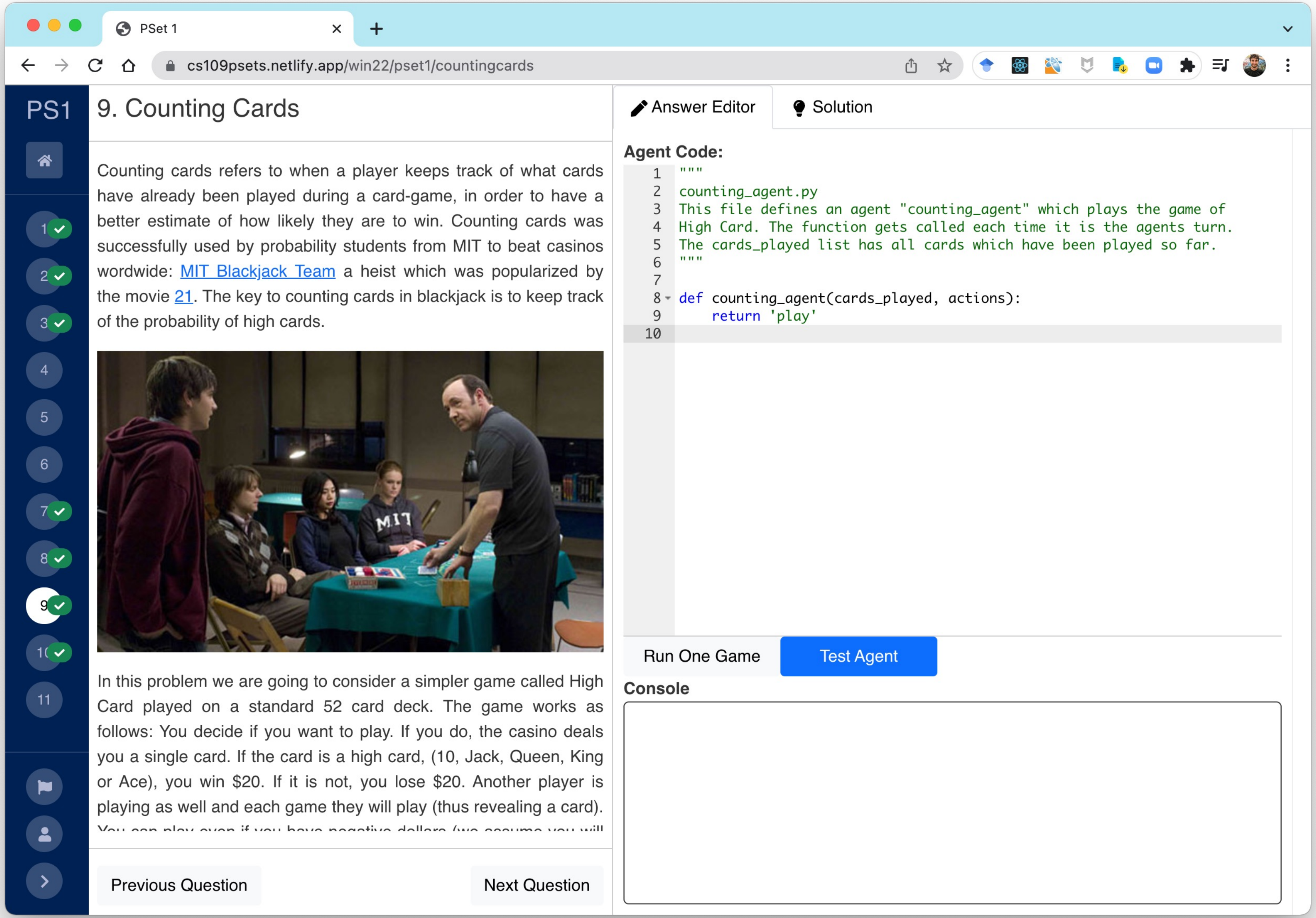
Problem Set #1 is out newish



Check your answer

Auto Submission

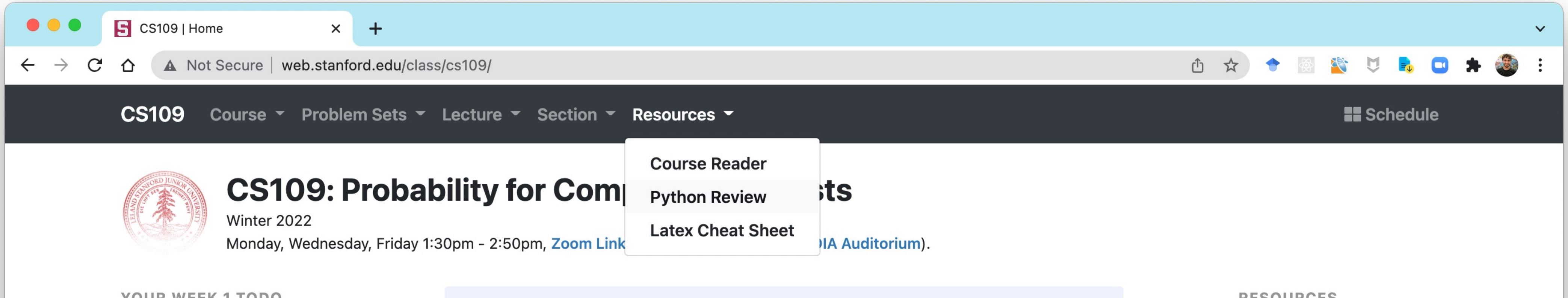
Write an Agent newish



Python Review Session

Friday at 5pm PT with Ishira (online)

Find links, recordings, and setup here



Learn LaTex

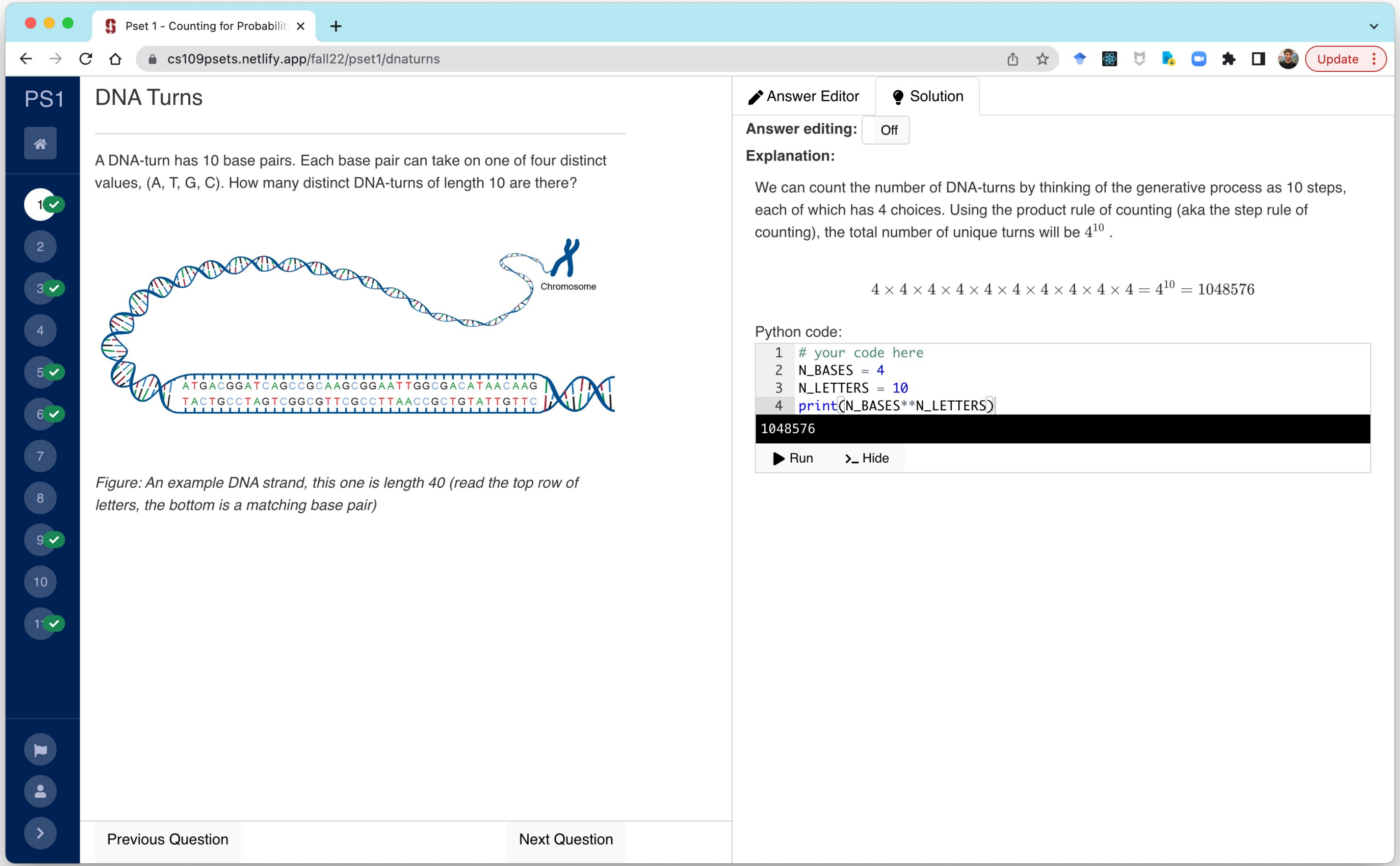
Making a handout to help you get started

```
\begin{aligned}
P(E) 
&= \sum_{i=0}^n e^{i}\\
&= 0.25
\end{aligned}
```

$$\begin{aligned} P(E) &= \sum_{i=0}^n e^{i}\\ &= 0.25 \end{aligned}$$

Done

What Makes for a Good Answer?



Honor Code

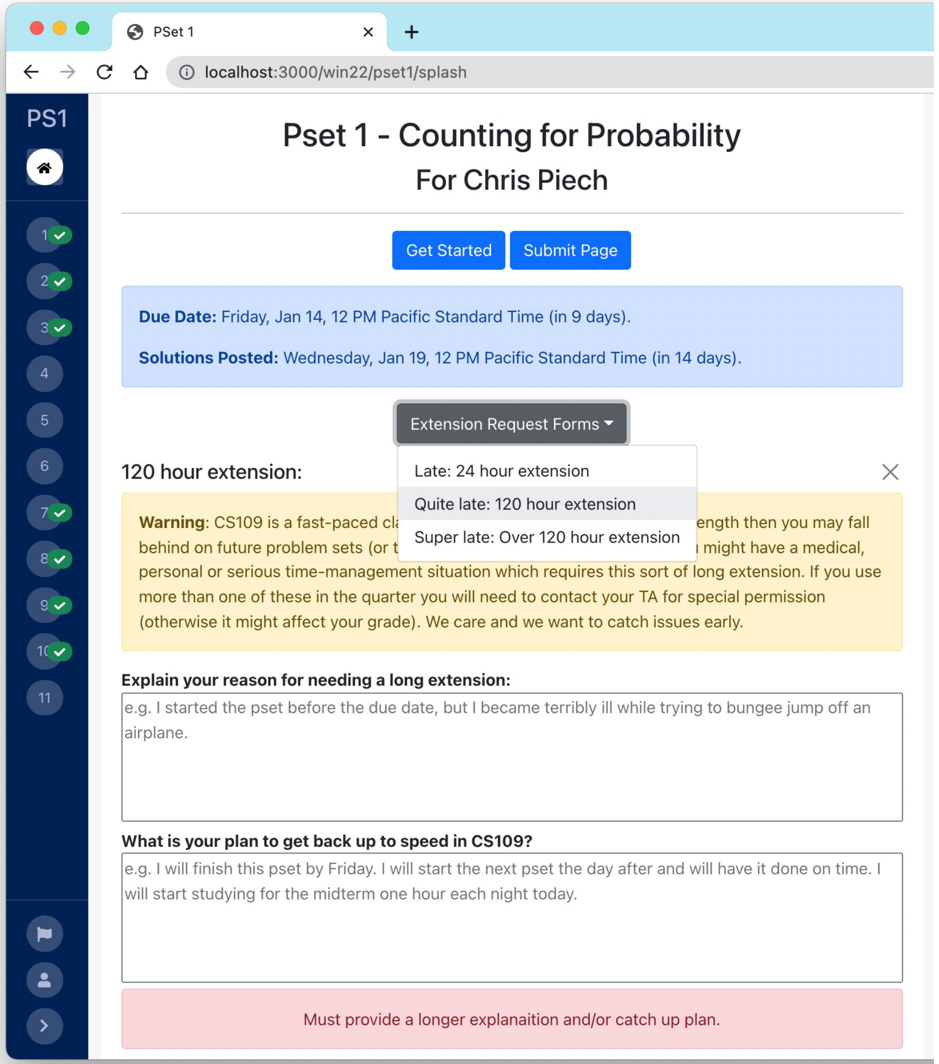
Always remember: You need to be able to recreate your ability on an exam. And in the real world. This is a foundation course. **Cheating in CS109 is cheating yourself**.

Talk to your friends about the **concepts**, not the solution. Words must be your own.

Practice the **art of teaching**. Three most important things to know:

1. Do not give away the answer
2. Always be respectful
3. Know what you don't know

Late Policy



Three types of extensions:

1. Grace period (24 hours)
2. Quite Late extension (2 class days)
3. After the **hard** deadline (> 2 class days)

You give them to yourself. Need to talk to us if:

A. You need more than 2 long extensions / qtr
B. Extension past the hard deadline

But CS109 is a fast class. If you want a long extension I want you to be intentional about how you are going to catch up.

If you notice a bug?

It should be robust, but things can happen.

Let me know: send an email to cpiech@stanford.edu or message me on slack. I need your email and the approximate time you encountered the bug.

Want to try to hack the PsetApp? We will give you time after class :-)


Above
&Beyond

Review

CS109: From Counting to Machine Learning

Counting Theory

Core Probability

Random Variables

Probabilistic Models

Uncertainty Theory

Machine Learning

Core Counting

Counting with steps

Definition: Step Rule of Counting (aka Product Rule of Counting)

If an experiment has two parts, where the first part can result in one of m outcomes and the second part can result in one of n outcomes regardless of the outcome of the first part, then the total number of outcomes for the experiment is $m \cdot n$.

Counting with "or"

Definition: Inclusion Exclusion Counting

If the outcome of an experiment can either be drawn from set A or set B, and sets A and B may potentially overlap (i.e., it is not the case that A and B are mutually exclusive), then the number of outcomes of the experiment is $|A \text{ or } B| = |A| + |B| - |A \text{ and } B|$.

How Many Bit Stings?

Problem: A 6-bit string is sent over a network. The valid set of strings recognized by the receiver must either start with "01" or end with "10". How many such strings are there?

Answer

$$N = |A| + |B| - |A \text{ and } B|$$
$$= 16 + 16 - 4$$
$$= 28$$

2^4 start with 01	2^4 end with 10
010000	000010
010001	000110
010010	001010
010011	001110
010100	**010010**
010101	**010110**
010110	**011010**
010111	**011110**
011000	100010
011001	100110
011010	101010
011011	101110
011100	110010
011101	110110
011110	111010
011111	111110
Set A	Set B

Challenge Problem

1. Strings
   - How many *different* orderings of letters are possible for the string BOBA?

BOBA, ABOB, OBBA…

End Review

Permutations I

Orderings of Letters

How many letter orderings are possible for the following strings?

1. CHRIS

This is Jerry's dog, Doris. She puts her little Doris paw up to her chin when she's thinking.

Orderings of Letters

chirs	crish	hicrs	hsirc	irchs	rcish	rschi	shirc
chisr	crshi	hicsr	hsrci	ircsh	rcshi	rscih	shrci
chris	crsih	hircs	hsric	irhcs	rcsih	rshci	shric
chrsi	cshir	hirsc	ichrs	irhsc	rhcis	rshic	sichr
chsir	cshri	hiscr	ichsr	irsch	rhcsi	rsich	sicrh
chsri	csihr	hisrc	icrhs	irshc	rhics	rsihc	sihcr
cihrs	csirh	hrcis	icrsh	ischr	rhisc	schir	sihrc
cihsr	csrhi	hrcsi	icshr	iscrh	rhsci	schri	sirch
cirhs	csrih	hrics	icsrh	ishcr	rhsic	scihr	sirhc
cirsh	hcirs	hrisc	ihcrs	ishrc	richs	scirh	srchi
cishr	hcisr	hrsci	ihcsr	isrch	ricsh	scrhi	srcih
cisrh	hcris	hrsic	ihrcs	isrhc	rihcs	scrih	srhci
crhis	hcrsi	hscir	ihrsc	rchis	rihsc	shcir	srhic
crhsi	hcsir	hscri	ihscr	rchsi	risch	shcri	srich
crihs	hcsri	hsicr	ihsrc	rcihs	rishc	shicr	srihc

Orderings of letters

Step 1:
Chose first letter

Step 2:
Chose 2nd letter

Step 3:
Chose 3rd letter

Step 4:
Chose 4th letter

Step 5:
Chose 5th letter

Orderings of letters

H C S I R

Step 1:
Chose first letter
(5 options)

Step 2:
Chose 2nd letter
(4 options)

Step 3:
Chose 3rd letter
(3 options)

Step 4:
Chose 4th letter
(2 options)

Step 5:
Chose 5th letter
(1 option)

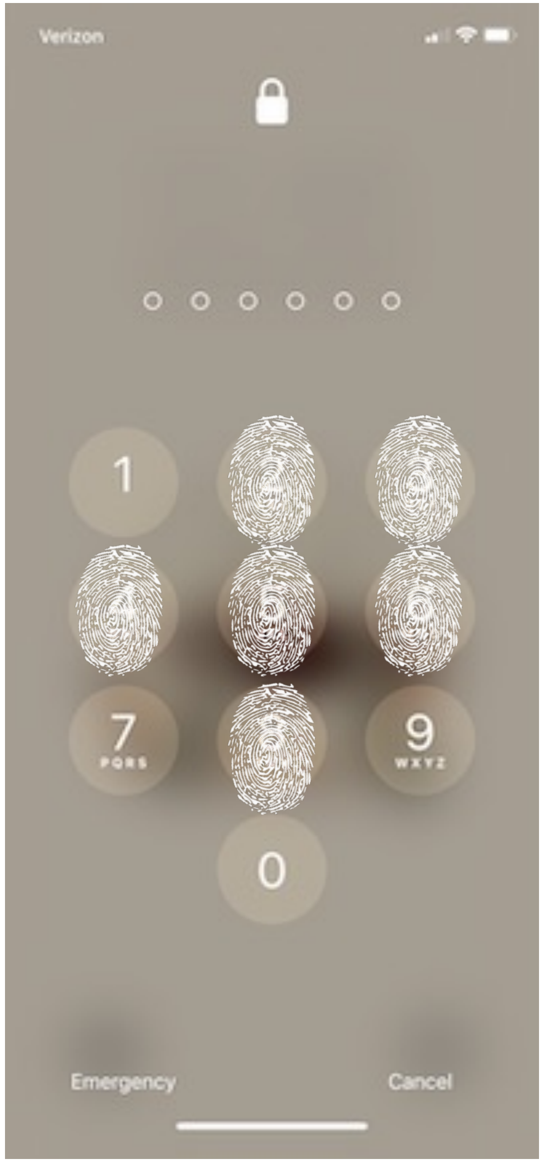
Permutations

A permutation is an ordered arrangement of objects.

The number of unique orderings (permutations) of n distinct objects is

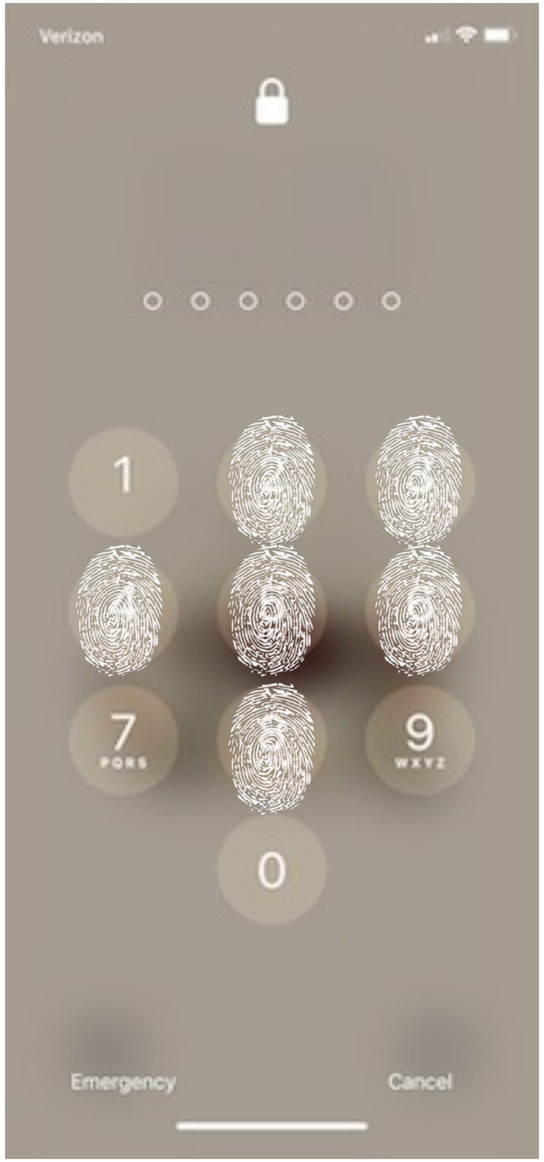
$$n! = n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1.$$

Unique 6-digit passcodes with six smudges



How many unique 6-digit passcodes are possible if a phone password uses each of **six** distinct numbers?

Unique 6-digit passcodes with six smudges



How many unique 6-digit passcodes are possible if a phone password uses each of **six** distinct numbers?

$$\text{Total} = 6! = 720 \text{ passcodes}$$

How many unique passcodes are possible if a phone password is some ordered subset of any 6 digits?

$$\text{Total} = 10 \times 9 \times 8 \times 7 \times 6 \times 5$$

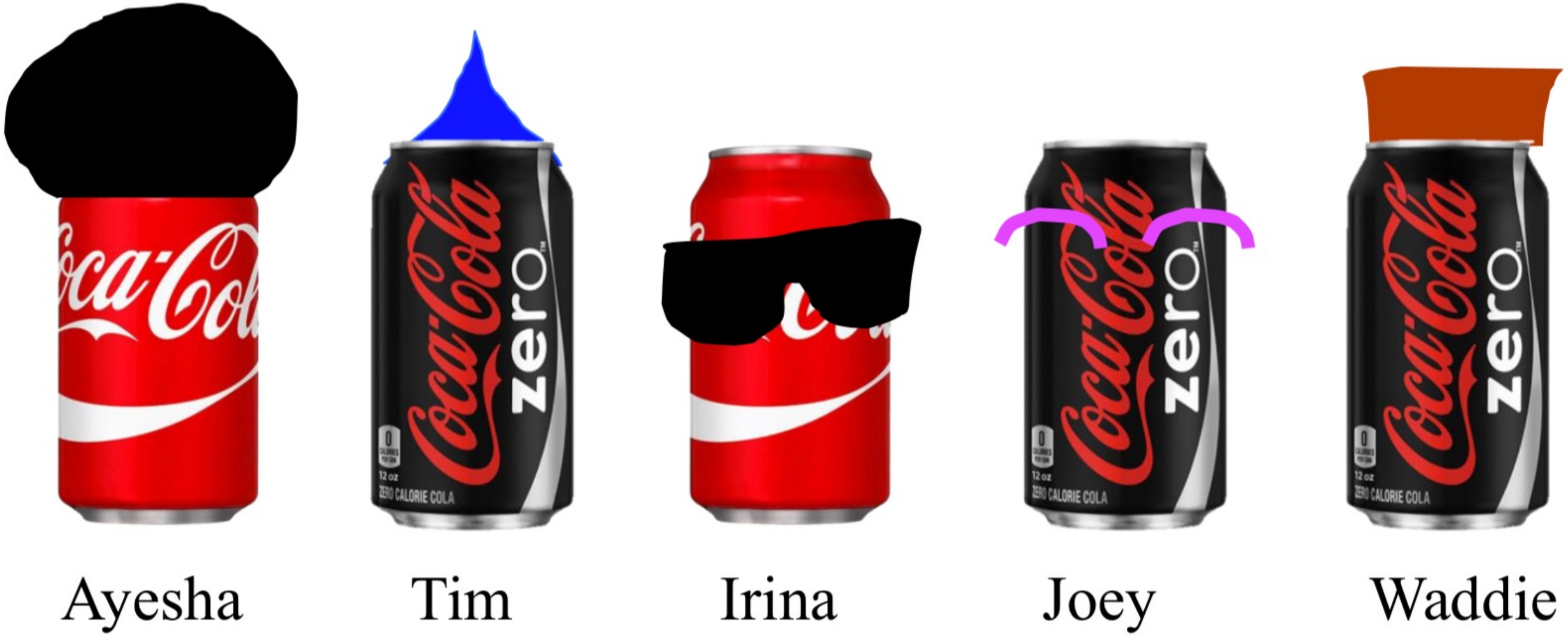
$$= \frac{10!}{4!} = 151200 \text{ passcodes}$$

Unique Bit Strings

1, 0, 1, 0, 0

Sort n distinct objects

Sort n distinct objects



Sort n distinct objects

1^{st} 2^{nd} 3^{rd} 4^{th} 5^{th}

<u>Steps</u>:

1. Choose 1^{st} can — 5 options
2. Choose 2^{nd} can — 4 options

…

5. Choose 5^{th} can — 1 option

Total $= 5 \times 4 \times 3 \times 2 \times 1$

$= 120$

Permutations II

Summary of Combinatorics

Counting tasks on n objects

Sort objects (permutations)

Choose k objects (combinations)

Put objects in r buckets

Distinct (distinguishable)

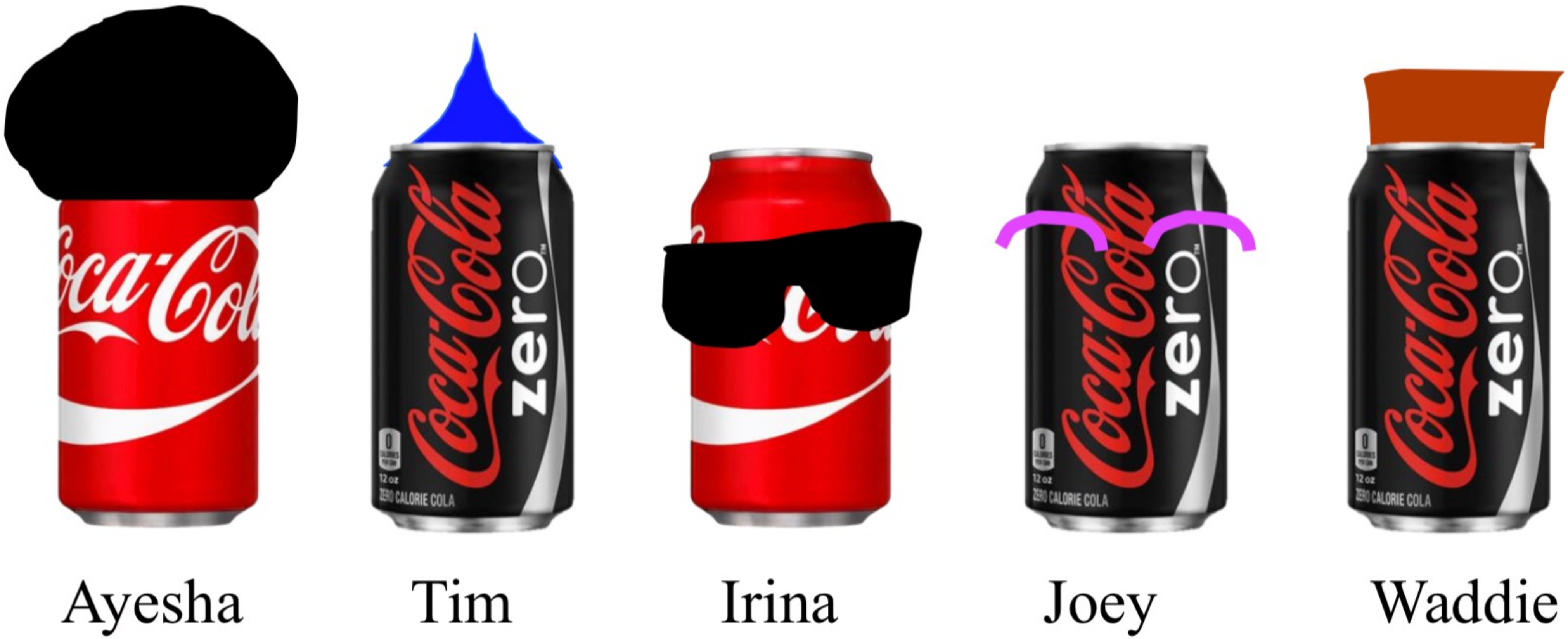
Summary of Combinatorics

Counting tasks on n objects

- Sort objects (permutations)
  - Distinct (distinguishable): $n!$
  - Some distinct
- Choose k objects (combinations)
- Put objects in r buckets

How many ways can we sort coke cans!

Sort n distinct objects

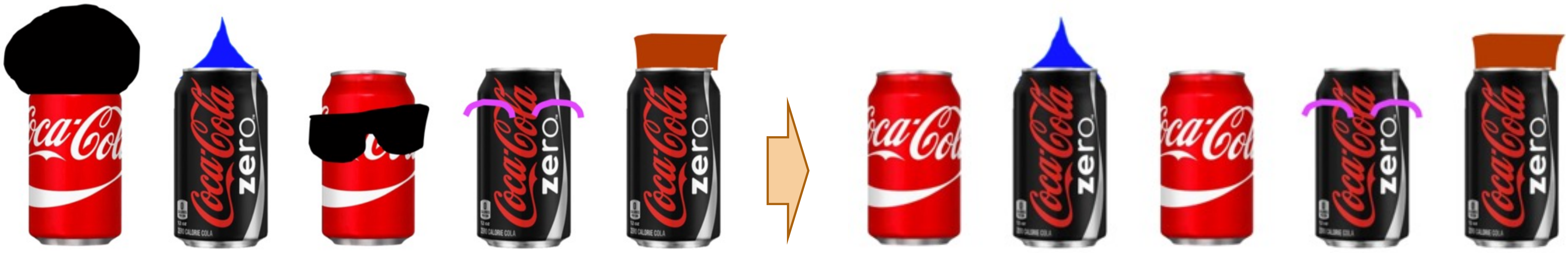


# of permutations =

Sort semi-distinct objects

Order n distinct objects	$n!$

All distinct　　　　Some indistinct



Ayesha　Tim　Irina　Joey　Waddie　→　Coke　Tim　Coke　Joey　Waddie

Sort semi-distinct objects

How do we find the number of permutations considering some objects are indistinct?

By the product rule, permutations of distinct objects is a two-step process:

permutations of distinct objects $=$ permutations considering some objects are indistinct $\times$ Permutations of just the indistinct objects

Sort semi-distinct objects

How do we find the number of permutations considering some objects are indistinct?

By the product rule, permutations of distinct objects is a two-step process:

$$\frac{\text{permutations of distinct objects}}{\text{Permutations of just the indistinct objects}} = \text{permutations considering some objects are indistinct}$$

General approach to counting permutations

When there are n objects such that

n_1 are the same (indistinguishable or indistinct), and

n_2 are the same, and

…

n_r are the same,

The number of unique orderings (permutations) is

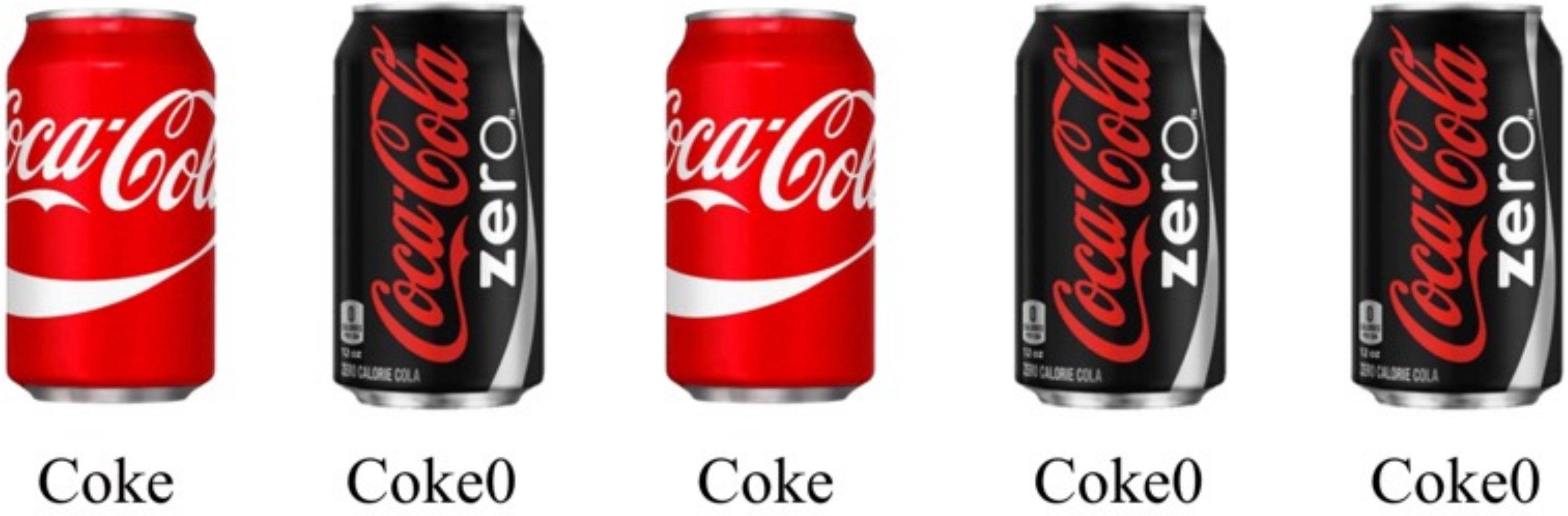
$$\frac{n!}{n_1!\,n_2!\cdots n_r!}.$$

For each group of indistinct objects,
Divide by the overcounted permutations.

Sort semi-distinct objects

Order n semi-distinct objects $\frac{n!}{n_1!\,n_2!\cdots n_r!}$

How many permutations?



Strings

Order n semi-distinct objects $\frac{n!}{n_1!\,n_2!\cdots n_r!}$

How many letter orderings are possible for the following strings?

1. BOBA

2. MISSISSIPPI

This is Jerry's dog, Doris. She puts her little Doris paw up to her chin when she's thinking.

Strings

Order n semi-distinct objects $\frac{n!}{n_1!\,n_2!\cdots n_r!}$

How many letter orderings are possible for the following strings?

1. BOBA $= \frac{4!}{2!} = 12$

2. MISSISSIPPI $= \frac{11!}{1!4!4!2!} = 34{,}650$

To the Code!

baob
bbao
obba
oabb
boab
babo
abbo
aobb
boba
abob
bboa
obab

```
import itertools

def main():
   letters = ['b','o','b','a']
   perms = set(itertools.permutations(letters))
   for perm in perms:
      pretty_perm = "".join(perm)
      print(pretty_perm)
```

```
import math

def main():
   n = math.factorial(4)
   d = math.factorial(2)
   print(n / d)
```

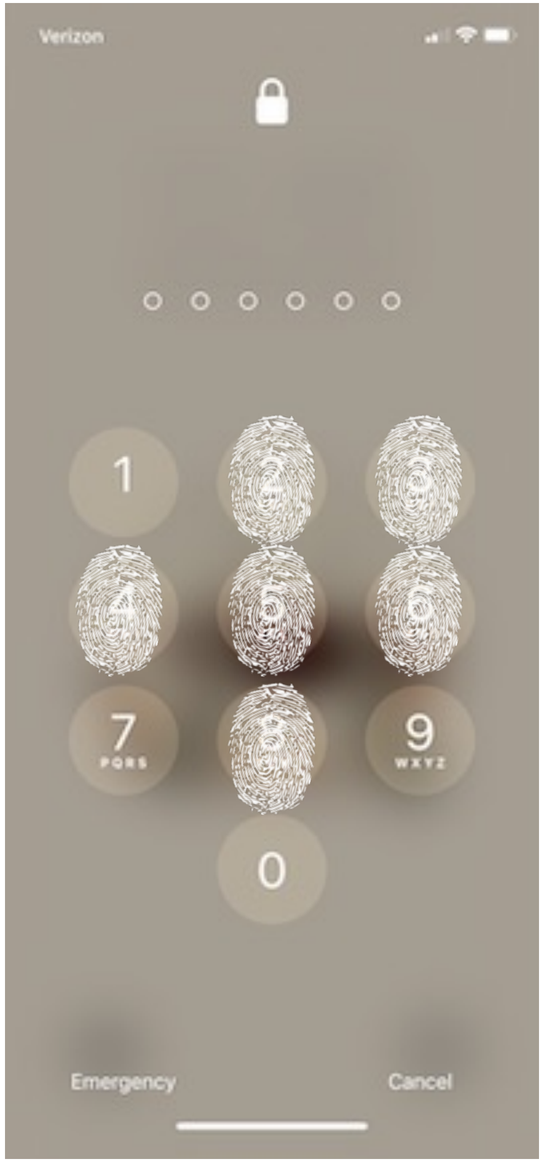
Summary of Combinatorics

Counting tasks on n objects

- Sort objects (permutations)
  - Distinct (distinguishable): $n!$
  - Some distinct: $\frac{n!}{n_1!\,n_2!\cdots n_r!}$
- Choose k objects (combinations)
- Put objects in r buckets

Unique 6-digit passcodes with six smudges

Order n semi-distinct objects $\frac{n!}{n_1!\,n_2!\cdots n_r!}$

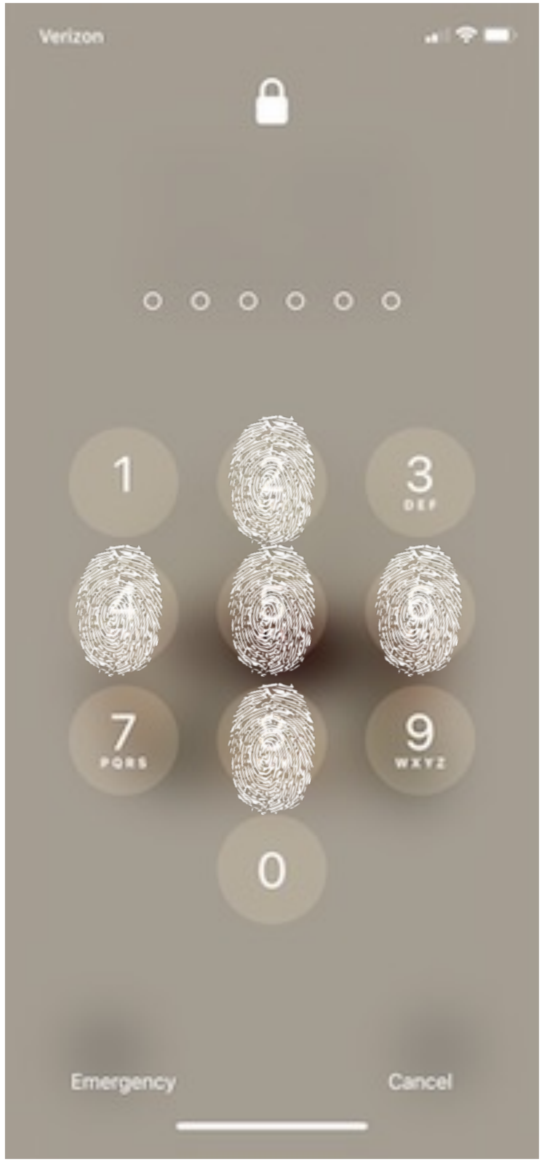


How many unique 6-digit passcodes are possible if a phone password uses each of six distinct numbers?

$$\text{Total} = 6!$$
$$= 720 \text{ passcodes}$$

Unique 6-digit passcodes with five smudges

Order n semi-distinct objects $\frac{n!}{n_1!\,n_2!\cdots n_r!}$



How many unique 6-digit passcodes are possible if a phone password uses each of five distinct numbers?

Steps:

1. Choose digit to repeat — 5 outcomes
2. Create passcode — (sort 6 digits: 4 distinct, 2 indistinct)

Total $= 5 \times \frac{6!}{2!}$

$= 1{,}800$ passcodes

Combinations I

Summary of Combinatorics

Counting tasks on n objects

- Sort objects (permutations)
  - Distinct (distinguishable): $n!$
  - Some distinct: $$\frac{n!}{n_1!\,n_2!\cdots n_r!}$$
- Choose k objects (combinations)
  - Distinct
- Put objects in r buckets

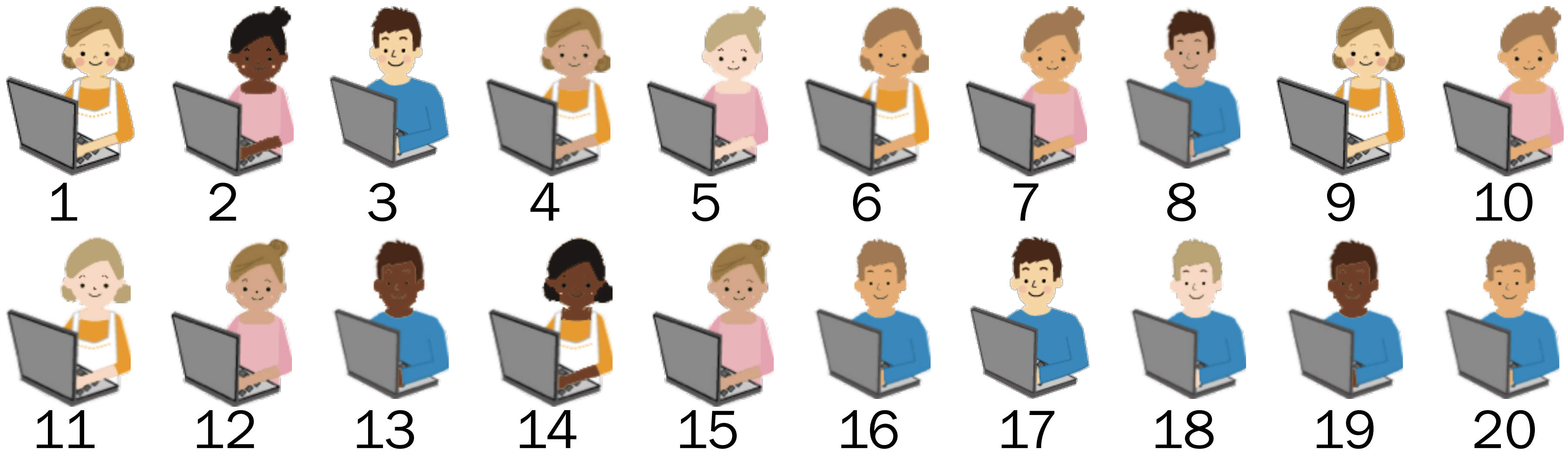
Combinations with cake

There are $n = 20$ people.
How many ways can we choose $k = 5$ people to get cake?

Consider the following generative process...

Combinations with cake

There are $n = 20$ people.
How many ways can we choose $k = 5$ people to get cake?

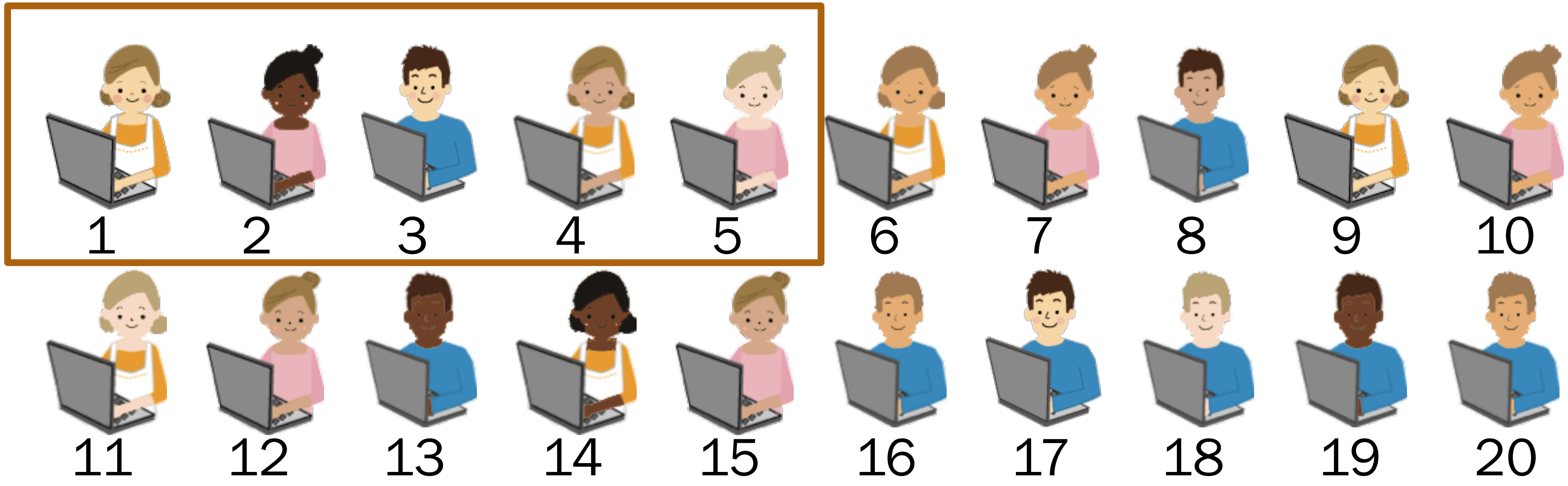


1. n people get in line

$n!$ ways

Combinations with cake

There are $n = 20$ people.
How many ways can we choose $k = 5$ people to get cake?



1. n people get in line

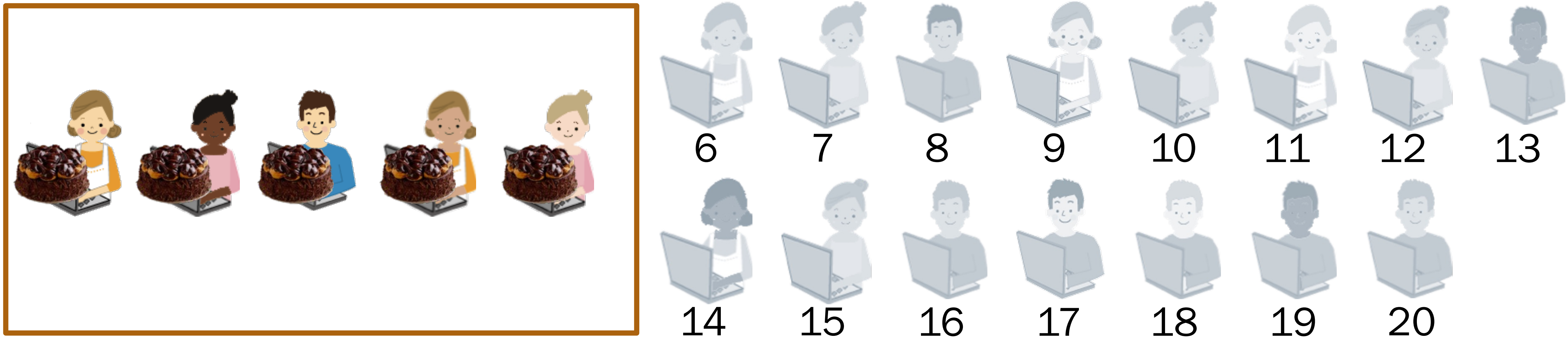
   $n!$ ways

2. Put first k in cake room

   1 way

Combinations with cake

There are $n = 20$ people.
How many ways can we choose $k = 5$ people to get cake?



1. n people get in line

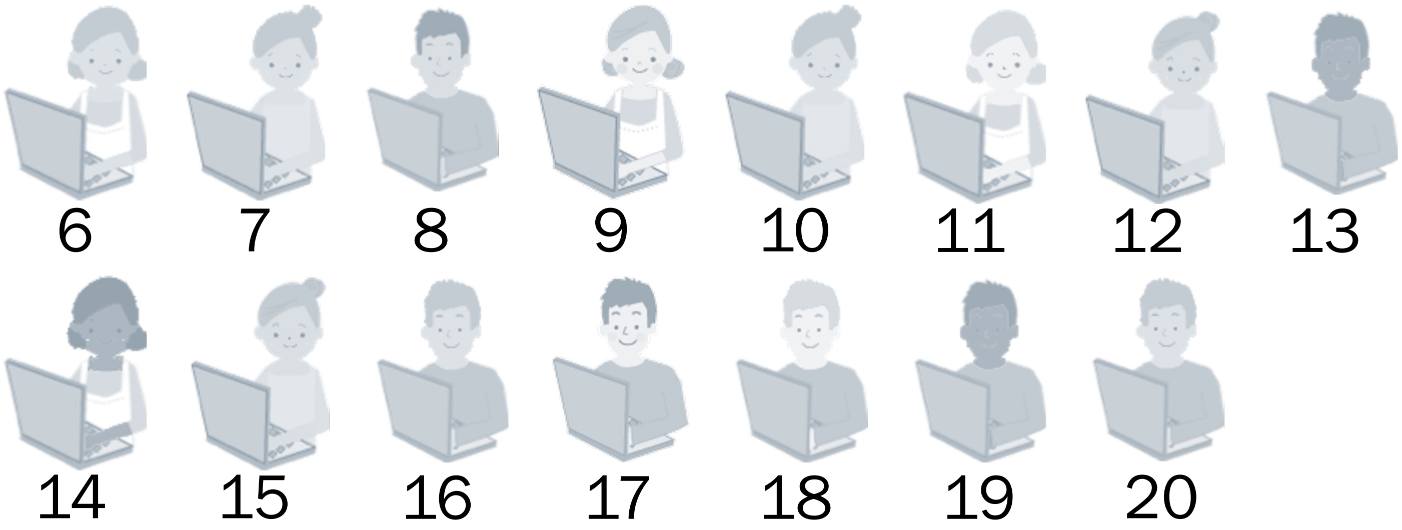
   $n!$ ways

2. Put first k in cake room

   1 way

Combinations with cake

There are $n = 20$ people.
How many ways can we choose $k = 5$ people to get cake?



1. n people get in line

   $n!$ ways

2. Put first k in cake room

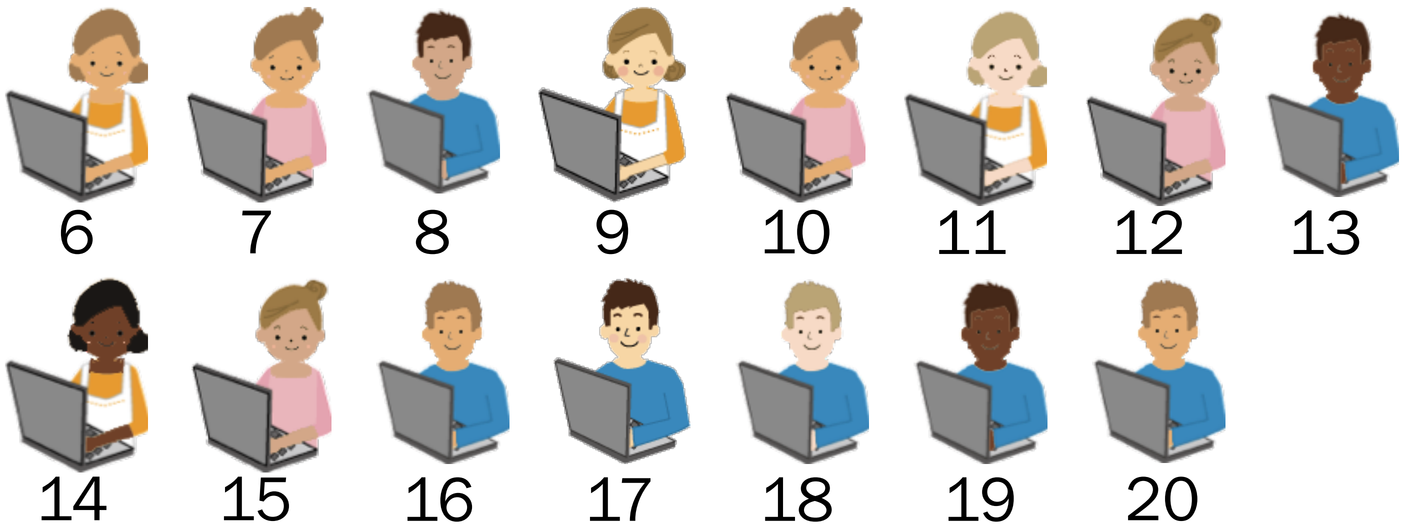
   1 way

3. Allow cake group to mingle

   $k!$ different permutations lead to the same mingle

Combinations with cake

There are $n = 20$ people.
How many ways can we choose $k = 5$ people to get cake?



1. n people get in line

   $n!$ ways

2. Put first k in cake room

   1 way

3. Allow cake group to mingle

   $k!$ different permutations lead to the same mingle

4. Allow non-cake group to mingle

Combinations with cake

There are $n = 20$ people.
How many ways can we choose $k = 5$ people to get cake?

1. n people get in line

   $n!$ ways

2. Put first k in cake room

   1 way

3. Allow cake group to mingle

   $k!$ different permutations lead to the same mingle

4. Allow non-cake group to mingle

   $(n-k)!$ different permutations lead to the same mingle

Combinations

A combination is an unordered selection of k objects from a set of n distinct objects.

The number of ways of making this selection is

$$\frac{n!}{k!\,(n-k)!} = n! \times 1 \times \frac{1}{k!} \times \frac{1}{(n-k)!}$$

1. Order n distinct objects
2. Take first k as chosen
3. Overcounted: any ordering of chosen group is same choice
4. Overcounted: any ordering of unchosen group is same choice

Combinations

A combination is an unordered selection of k objects from a set of n distinct objects.

The number of ways of making this selection is

$$\frac{n!}{k!\,(n-k)!} = n! \times 1 \times \frac{1}{k!} \times \frac{1}{(n-k)!} = \binom{n}{k}$$

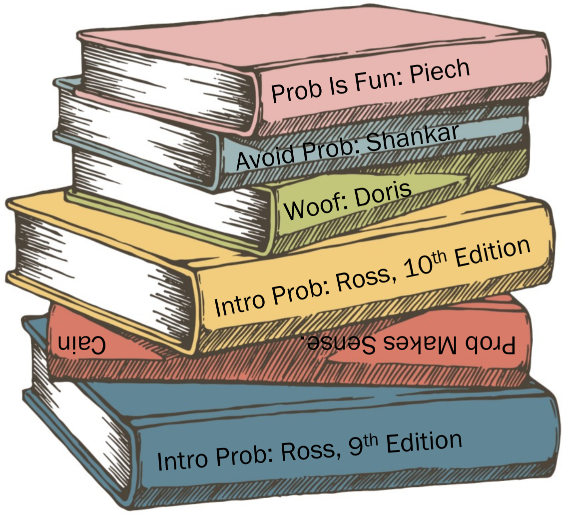
Binomial coefficient

Fun Fact: $\binom{n}{k} = \binom{n}{n-k}$

Probability textbooks

Choose k of n distinct objects $\binom{n}{k}$

How many ways are there to choose 3 books from a set of 6 distinct books?



$$\binom{6}{3} = \frac{6!}{3!\,3!} = 20 \text{ ways}$$

To the code!

How many unique hands of 5 cards are there in a 52 card deck?


```
def main():
    cards = make_deck()
    all_hands = itertools.combinations(cards, 5)
    for hand in all_hands:
        print(hand)
```

```
def main():
    total = math.comb(52, 5)
    print(total)
```

Buckets and The Divider Method

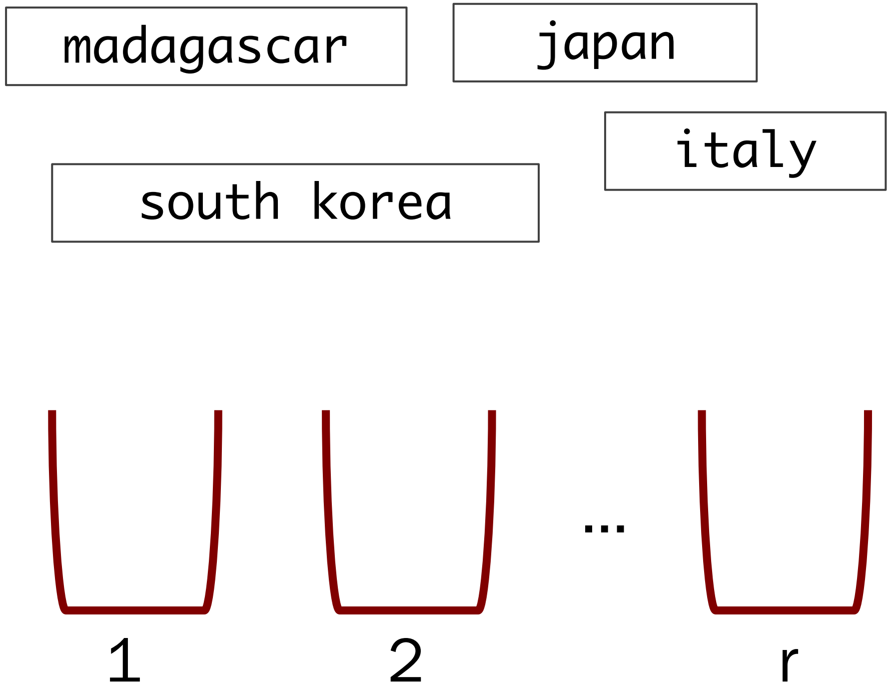
Summary of Combinatorics

Counting tasks on n objects

- **Sort objects (permutations)**
  - Distinct (distinguishable): $n!$
  - Some distinct: $\frac{n!}{n_1!\,n_2!\cdots n_r!}$
- **Choose k objects (combinations)**
  - Distinct: $\binom{n}{k}$
- **Put objects in r buckets**
  - Distinct
  - Indistinct

~~Balls and urns~~ Hash tables and distinct strings

How many ways are there to hash n distinct strings to r buckets?



Steps:

1. Bucket 1st string
2. Bucket 2nd string

...

n. Bucket nth string

r^n outcomes

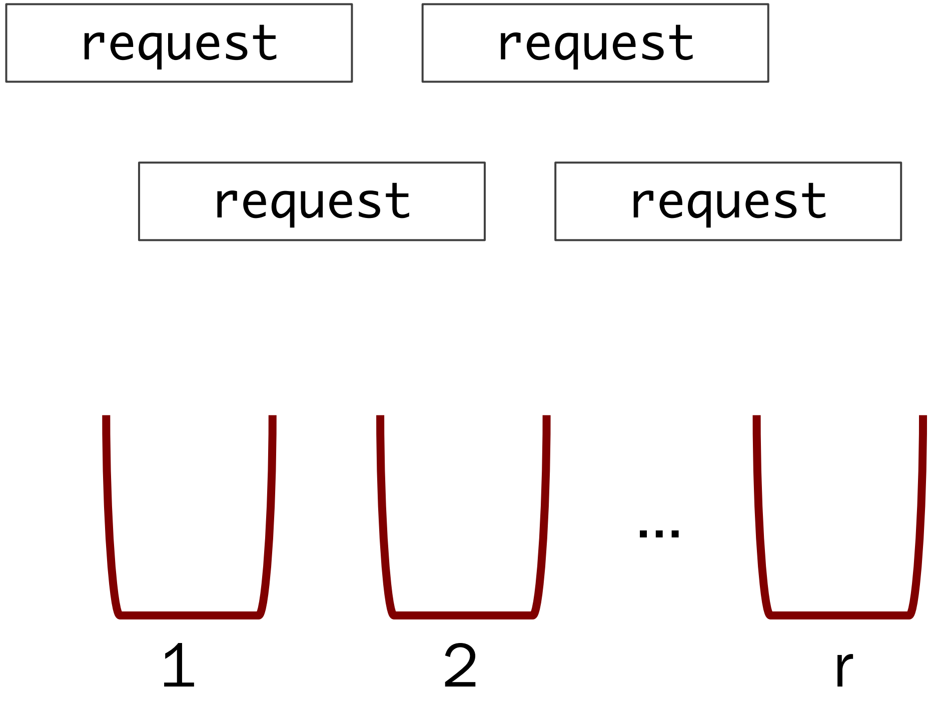
Summary of Combinatorics

Counting tasks on n objects

- Sort objects (permutations)
  - Distinct (distinguishable): $n!$
  - Some distinct: $\frac{n!}{n_1!\,n_2!\cdots n_r!}$
- Choose k objects (combinations)
  - Distinct: $\binom{n}{k}$
- Put objects in r buckets
  - Distinct: r^n
  - Indistinct

Servers and indistinct requests

How many ways are there to distribute n indistinct web requests to r servers?



Goal

Server 1 has x_1 requests,

Server 2 has x_2 requests,

...

Server r has x_r requests

constraint: $\sum_{i=1}^{r} x_i = n$

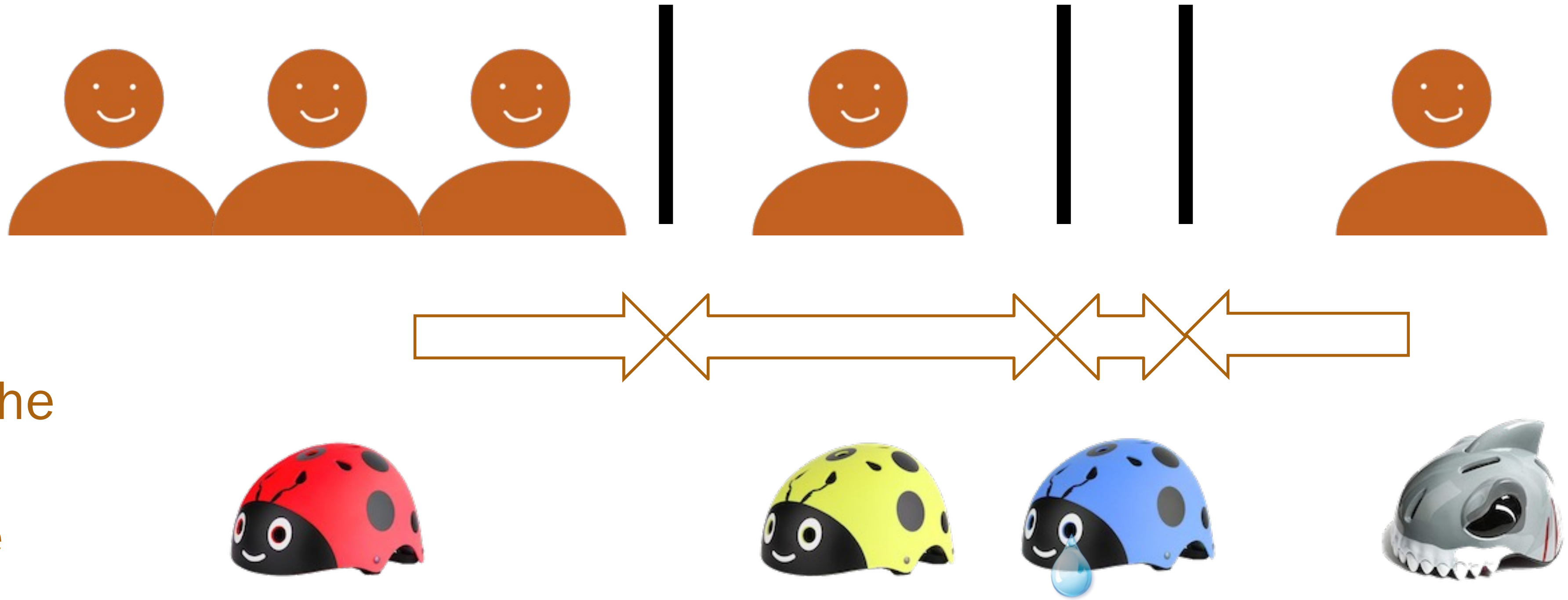
Bicycle helmet sales

How many ways can we assign $n = 5$ indistinct children to $r = 4$ distinct bicycle helmet styles?

Bicycle helmet sales

1 possible assignment outcome:

Goal Order n indistinct objects and $r - 1$ indistinct dividers.



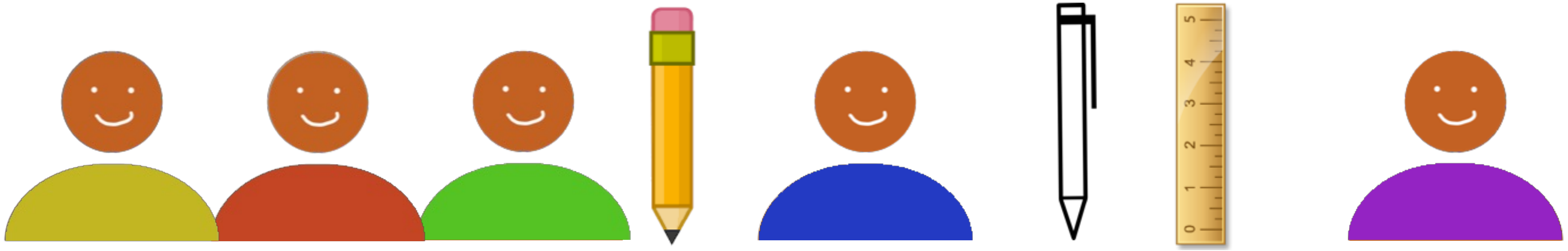
Consider the following generative process…

The divider method: A generative proof

How many ways can we assign $n = 5$ indistinct children to $r = 4$ distinct bicycle helmet styles?

Goal Order n indistinct objects and $r - 1$ indistinct dividers.

0. Make objects and dividers distinct

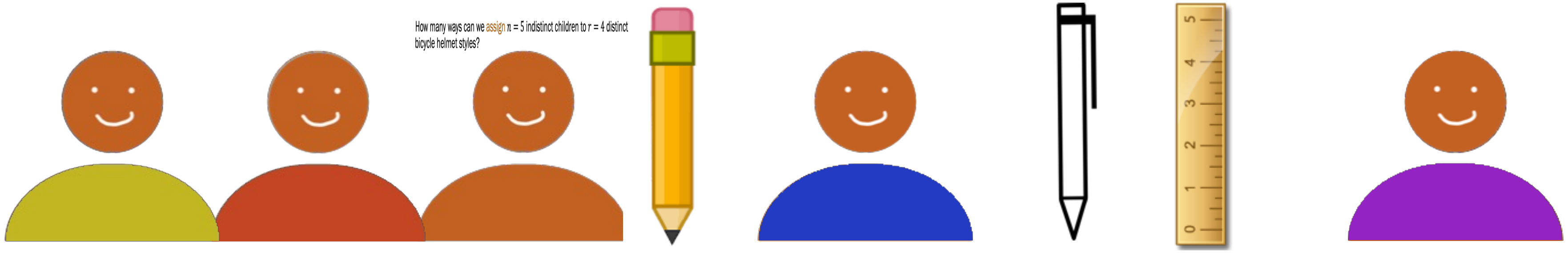


The divider method: A generative proof

How many ways can we assign $n = 5$ indistinct children to $r = 4$ distinct bicycle helmet styles?

Goal Order n indistinct objects and $r-1$ indistinct dividers.

0. Make objects and dividers distinct



1. Order n distinct objects and $r-1$ distinct dividers

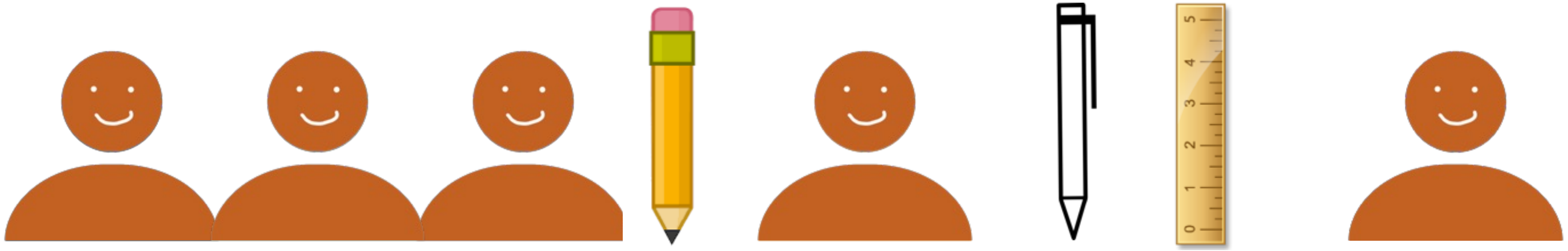
$$(n+r-1)!$$

The divider method: A generative proof

How many ways can we assign $n = 5$ indistinct children to $r = 4$ distinct bicycle helmet styles?

Goal Order n indistinct objects and $r - 1$ indistinct dividers.

0. Make objects and dividers distinct



1. Order n distinct objects and $r - 1$ distinct dividers

$$(n + r - 1)!$$

2. Make n objects indistinct

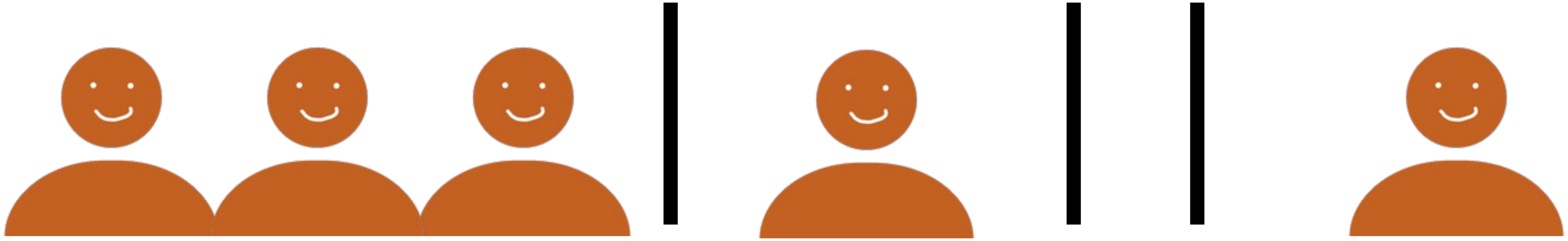
$$\frac{1}{n!}$$

The divider method: A generative proof

How many ways can we assign $n = 5$ indistinct children to $r = 4$ distinct bicycle helmet styles?

Goal Order n indistinct objects and $r - 1$ indistinct dividers.

0. Make objects and dividers distinct



1. Order n distinct objects and $r - 1$ distinct dividers

$$(n + r - 1)!$$

2. Make n objects indistinct

$$\frac{1}{n!}$$

3. Make $r - 1$ dividers indistinct

$$\frac{1}{(r-1)!}$$

The divider method

The number of ways to distribute n indistinct objects into r buckets is equivalent to the number of ways to permute $n + r - 1$ objects such that

n are indistinct objects, and

$r - 1$ are indistinct dividers:

$$\text{Total} = (n + r - 1)! \times \frac{1}{n!} \times \frac{1}{(r-1)!}$$

$$= \binom{n + r - 1}{r - 1} \text{ outcomes}$$

Integer solutions to equations

Divider method $\binom{n+r-1}{r-1}$
(n indistinct objects, r buckets)

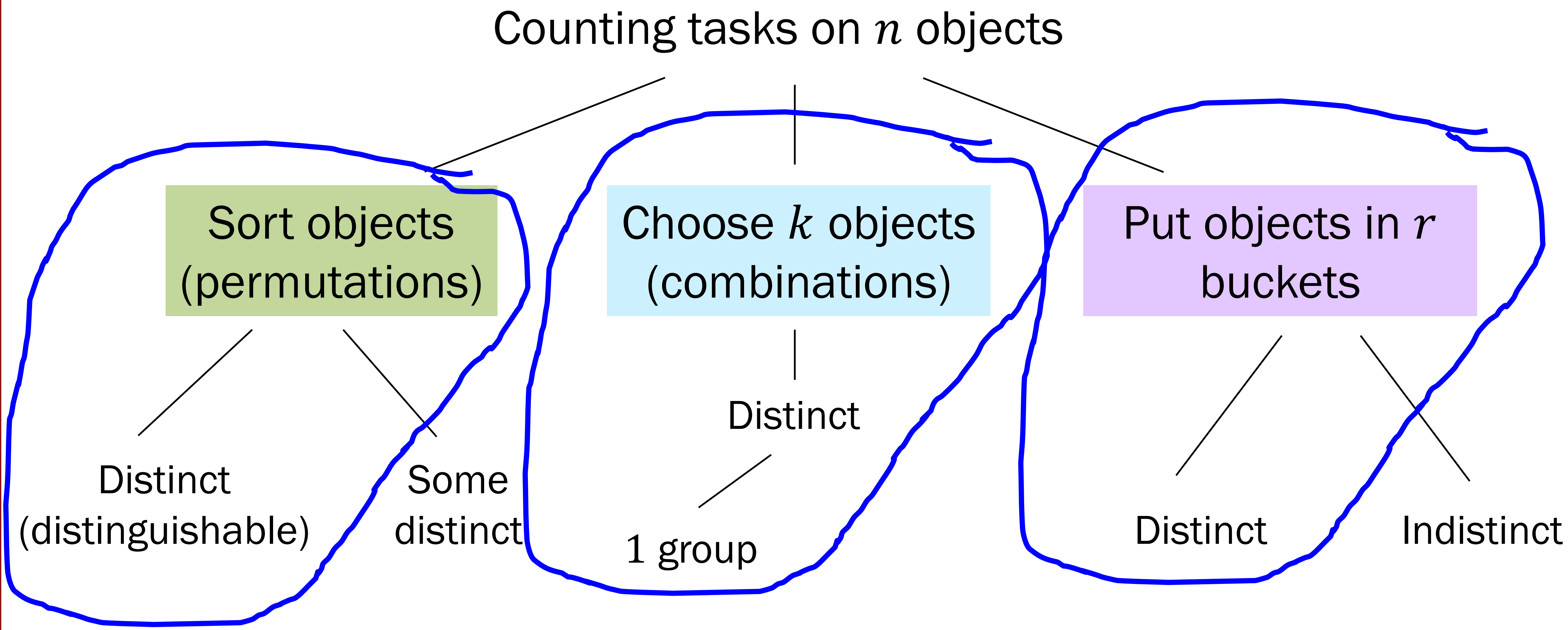
How many integer solutions are there to the following equation:

$$x_1 + x_2 + \cdots + x_r = n,$$

where for all i, x_i is an integer such that $0 \le x_i \le n$?

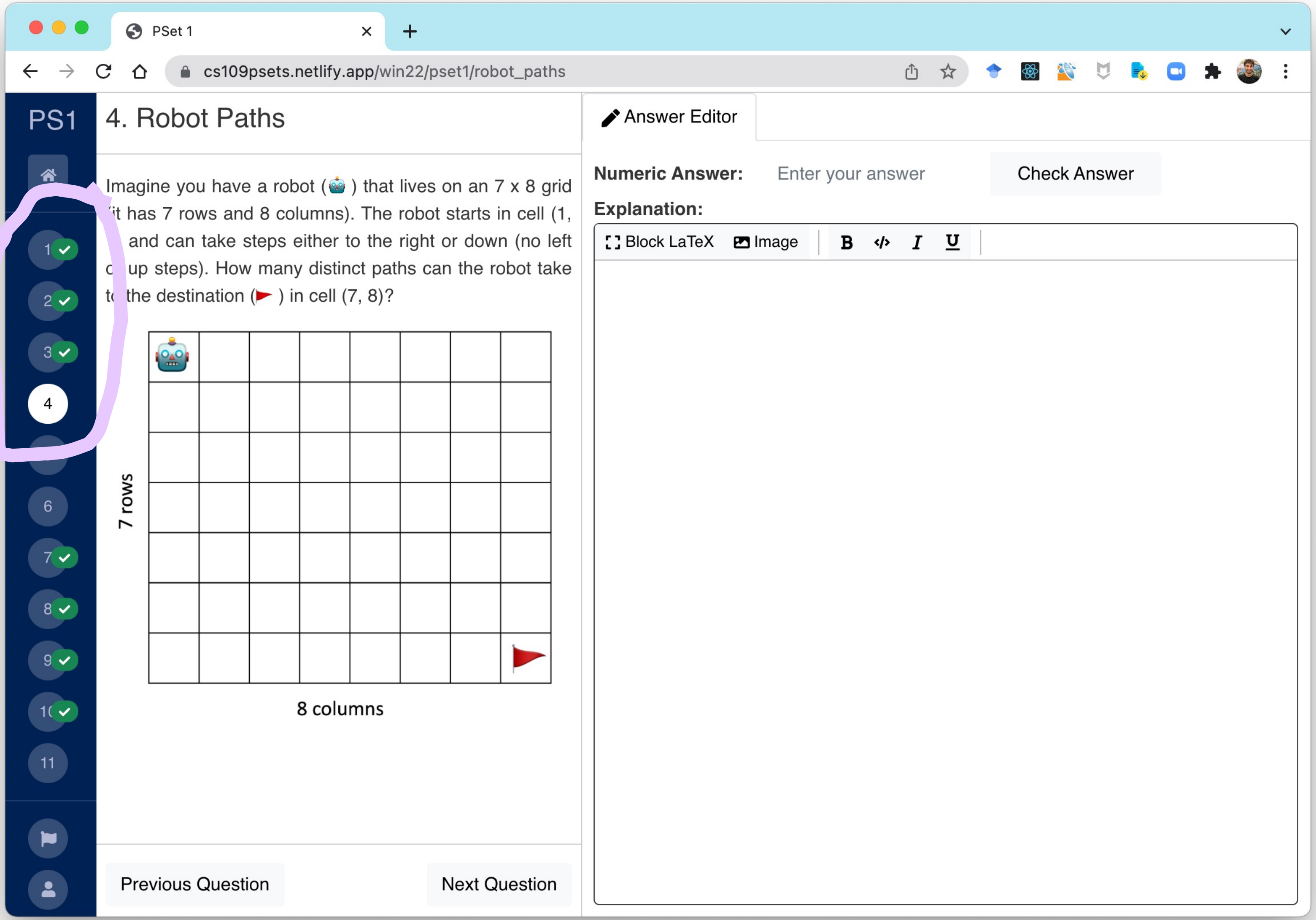
Positive integer equations can be solved with the divider method.

Summary of Combinatorics



Ready for the first 4 problems (and the rest on Friday)

Now, more than ever, get started early



Want to go deep?

Enigma Machine

chrispiech.github.io/probabilityForComputerScientists/en/examples/enigma/

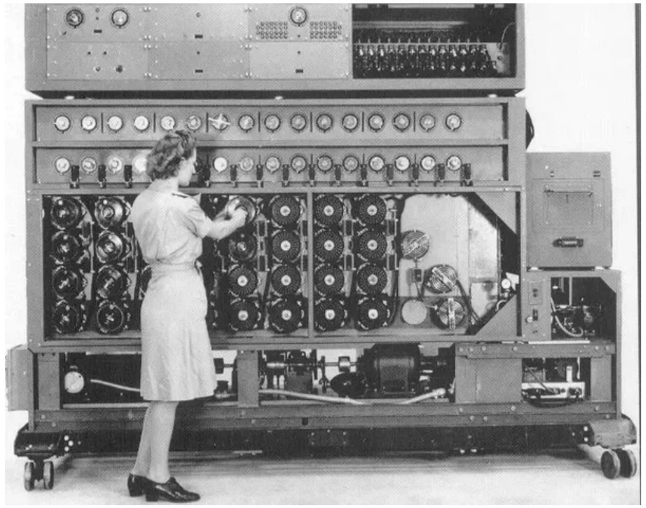
Update

Course Reader for CS109

Search book...

Enigma Machine

One of the very first computers was built to break the Nazi "enigma" codes in WW2. It was a hard problem because the "enigma" machine, used to make secret codes, had so many unique configurations. Every day the Nazis would choose a new configuration and if the Allies could figure out the daily configuration, they could read all enemy messages. One solution was to try all configurations until one produced legible German. This begs the question: How many configurations are there?



The WW2 machine built to search different enigma configurations.

The enigma machine has three rotors. Each rotor can be set to one of 26 different positions. How

Challenge: Bucketing Distincts into Many (Fixed Sized) Containers

You run an experiment that has three types of outcomes {A, B, C}. Among 10 runs you observe:

7 outcomes of type A

2 outcomes of type B

1 outcome of type C

There are 3^{10} unique orderings of outcomes. How many of those have exactly 7 As, 2 Bs and 3 Cs?