

**CS109: Probability for
Computer Scientists**

Problem Set #1 is out

newish

PS1 5. Random Choice

What is the probability that both users will get the same randomly generated password? Provide an answer to three decimal places!

```
import random

def main():
    user_1_password = generate_password()
    user_2_password = generate_password()

def generate_password():
    part_1 = random.choice([
        'red',
        'funky',
        'smelly'
    ])
    part_2 = random.choice([
        'apple',
        'pear',
        'pineapple'
    ])
    return part_1 + '-' + part_2
```

Answer Editor

Numeric Answer: 97

Check Answer

Explanation:

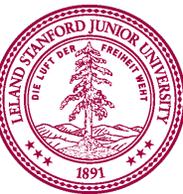
I am so excited! What a good time!

Insert LaTeX

Check your answer

Auto Submission

Previous Question Next Question



Write an Agent **newish**

CS109 PSet 1

9. Counting Cards

Counting cards refers to when a player keeps track of what cards have already been played during a card-game, in order to have a better estimate of how likely they are to win. Counting cards was successfully used by probability students from MIT to beat casinos worldwide: [MIT Blackjack Team](#) a heist which was popularized by the movie [21](#). The key to counting cards in blackjack is to keep track of the probability of high cards.



In this problem we are going to consider a simpler game called High Card played on a standard 52 card deck. The game works as follows: You decide if you want to play. If you do, the casino deals you a single card. If the card is a high card, (10, Jack, Queen, King or Ace), you win \$20. If it is not, you lose \$20. Another player is playing as well and each game they will play (thus revealing a card). You can play even if you have negative dollars (we assume you will

Answer Editor Solution

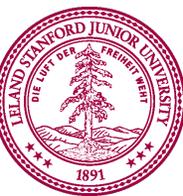
Agent Code:

```
1 """
2 counting_agent.py
3 This file defines an agent "counting_agent" which plays the game of
4 High Card. The function gets called each time it is the agents turn.
5 The cards_played list has all cards which have been played so far.
6 """
7
8 def counting_agent(cards_played, actions):
9     return 'play'
10
```

Run One Game Test Agent

Console

Previous Question Next Question



Python Review Session



Friday at 5pm PT with Ishira
(online)

Find links, recordings, and setup here

The screenshot shows a web browser window with the URL `web.stanford.edu/class/cs109/`. The page header includes 'CS109' and a navigation menu with 'Course', 'Problem Sets', 'Lecture', 'Section', and 'Resources'. The 'Resources' menu is open, showing 'Course Reader', 'Python Review', and 'Latex Cheat Sheet'. The main content area features the Stanford University logo, the course title 'CS109: Probability for Computer Scientists', the semester 'Winter 2022', and the schedule 'Monday, Wednesday, Friday 1:30pm - 2:50pm, Zoom Link (SIA Auditorium)'. At the bottom, there are sections for 'YOUR WEEK 1 TODO' and 'RESOURCES'.



Learn LaTeX

Making a handout to help you get started

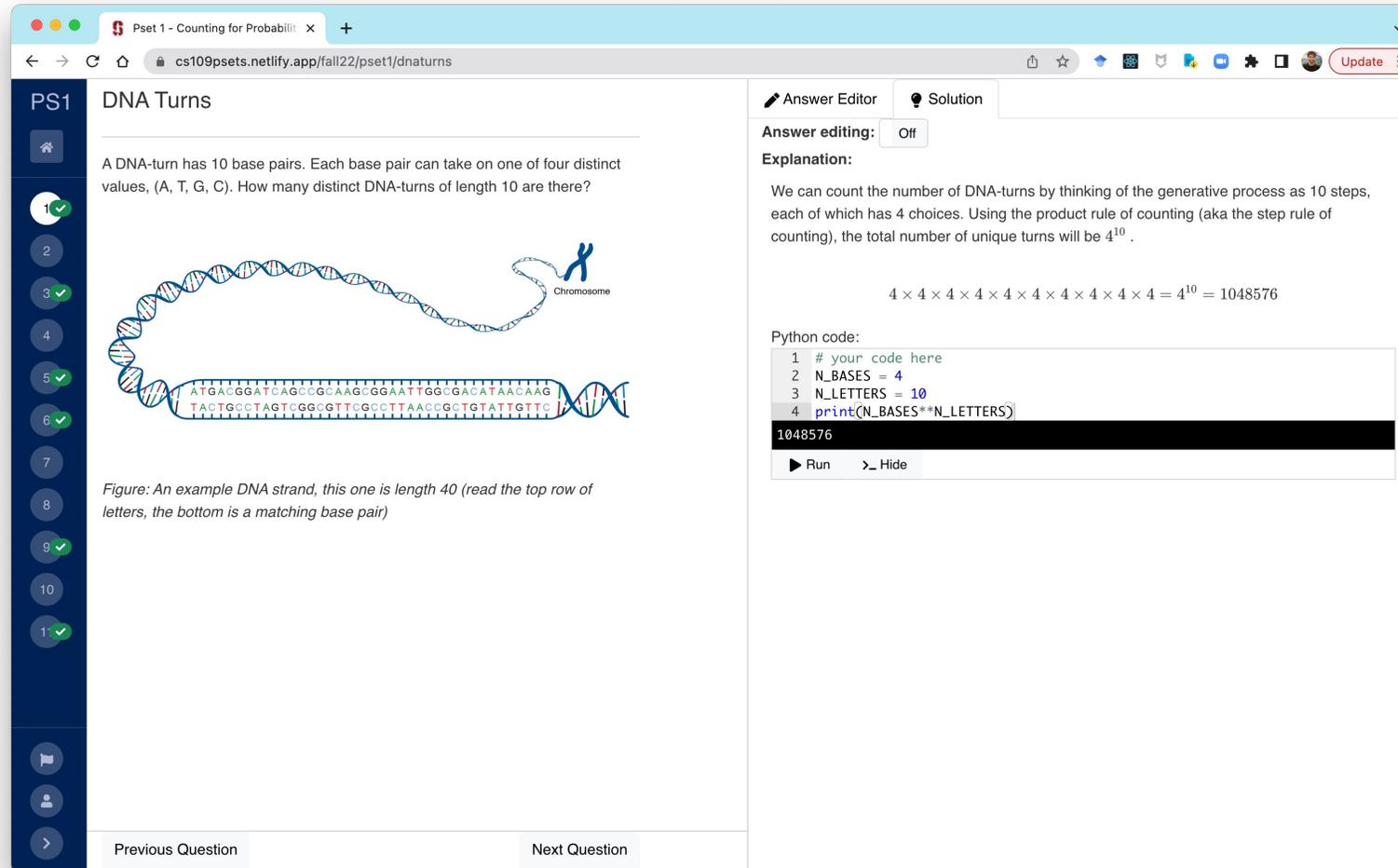
```
1 \begin{aligned}
2 P(E) |
3 &= \sum_{i=0}^n e^i \\
4 &= 0.25
5 \end{aligned}
```

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

Done



What Makes for a Good Answer?



The screenshot shows a web browser window with a problem set titled "PS1 DNA Turns". The question asks for the number of distinct DNA-turns of length 10, given that each base pair can be one of four values (A, T, G, C). The solution is provided in the "Solution" tab, explaining that the total number of unique turns is $4^{10} = 1048576$. The solution also includes a Python code snippet that calculates this value.

PS1 DNA Turns

A DNA-turn has 10 base pairs. Each base pair can take on one of four distinct values, (A, T, G, C). How many distinct DNA-turns of length 10 are there?

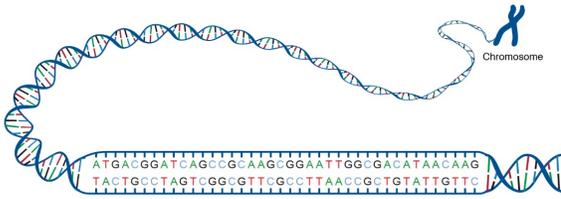


Figure: An example DNA strand, this one is length 40 (read the top row of letters, the bottom is a matching base pair)

Answer Editor **Solution**

Answer editing: Off

Explanation:

We can count the number of DNA-turns by thinking of the generative process as 10 steps, each of which has 4 choices. Using the product rule of counting (aka the step rule of counting), the total number of unique turns will be 4^{10} .

$$4 \times 4 = 4^{10} = 1048576$$

Python code:

```
1 # your code here
2 N_BASES = 4
3 N_LETTERS = 10
4 print(N_BASES**N_LETTERS)
```

1048576

▶ Run >_ Hide

Previous Question Next Question



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



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 Strings?

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

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

2^4 start with 01

010000
010001
010010
010011
010100
010101
010110
010111
011000
011001
011010
011011
011100
011101
011110
011111

Set A

2^4 end with 10

000010
000110
001010
001110
010010
010110
011010
011110
100010
100110
101010
101110
110010
110110
111010
111110

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	ihrcs	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

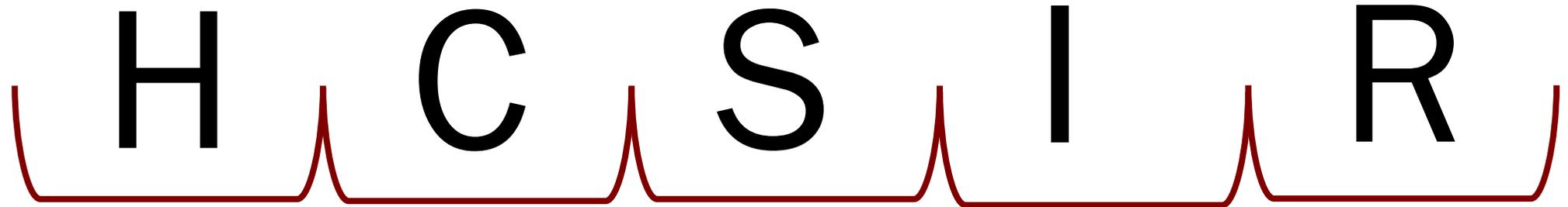
Step 2:
Chose 2nd letter

Step 3:
Chose 3rd letter

Step 4:
Chose 4th letter

Step 5:
Chose 5th letter

Orderings of letters



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)

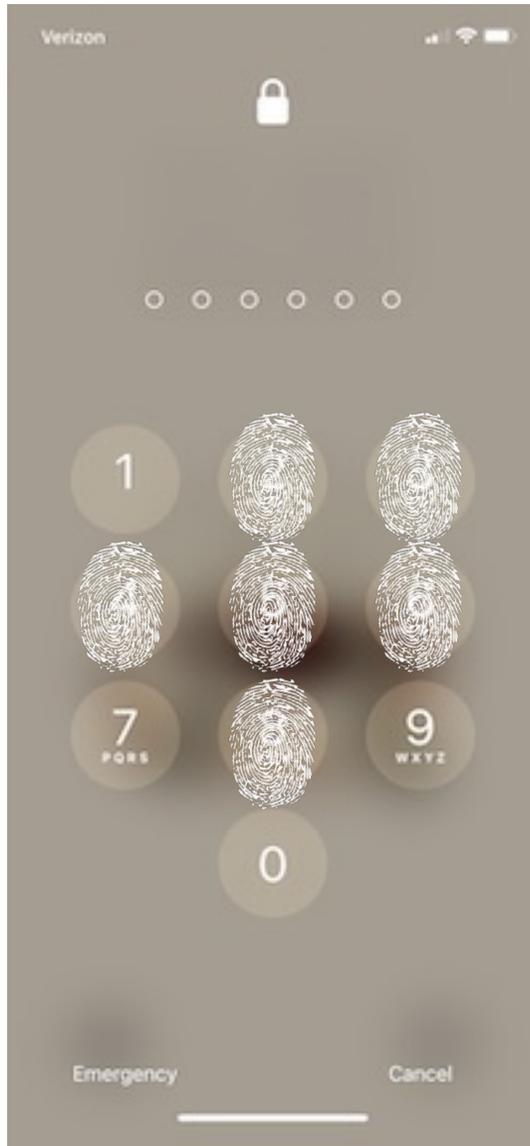
Permutations

A **permutation** is an ordered arrangement of objects.

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

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

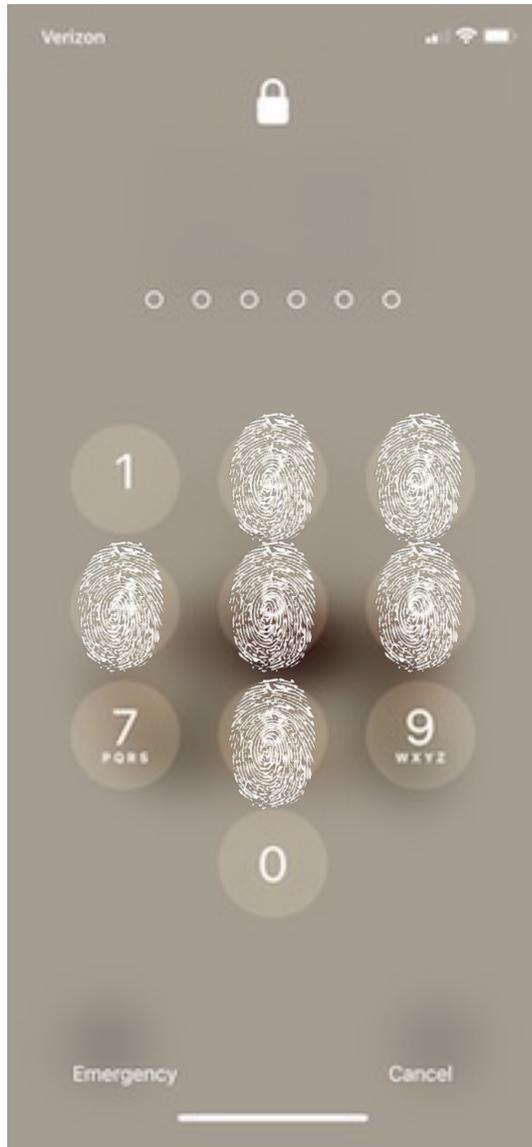
Unique 6-digit passcodes with **six** smudges



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?

$$\begin{aligned} \text{Total} &= 10 \times 9 \times 8 \times 7 \times 6 \times 5 \\ &= \frac{10!}{4!} = 151200 \text{ passcodes} \end{aligned}$$



Unique Bit Strings

1, 0, 1, 0, 0



Sort n distinct objects



Sort n distinct objects



Ayesha



Tim



Irina



Joey



Waddie

Sort n distinct objects



Steps:

1. Choose 1st can 5 options
2. Choose 2nd can 4 options
- ...
5. Choose 5th can 1 option

$$\begin{aligned} \text{Total} &= 5 \times 4 \times 3 \times 2 \times 1 \\ &= 120 \end{aligned}$$



Permutations II

Summary of Combinatorics

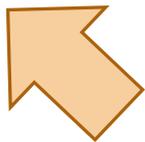
Counting tasks on n objects

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)

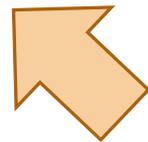
Choose k objects
(combinations)

Put objects in r
buckets

Distinct
(distinguishable)

Some
distinct

$n!$



How many ways can we sort coke cans!



Coke



Coke0



Coke



Coke0



Coke0

Sort n distinct objects



Ayesha



Tim



Irina



Joey



Waddie

of permutations =

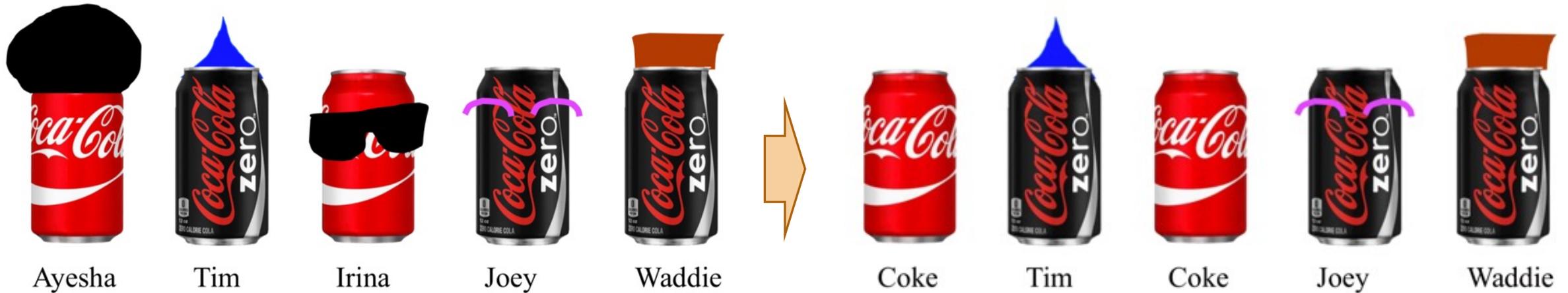
Sort semi-distinct objects

Order n
distinct objects

$n!$

All distinct

Some indistinct



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:

$$\begin{array}{l} \text{permutations} \\ \text{of distinct objects} \end{array} = \begin{array}{l} \text{permutations} \\ \text{considering some} \\ \text{objects are indistinct} \end{array} \times \begin{array}{l} \text{Permutations} \\ \text{of just the} \\ \text{indistinct objects} \end{array}$$

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

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

How many permutations?



Coke



Coke0



Coke



Coke0



Coke0

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.



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)

Choose k objects
(combinations)

Put objects in r
buckets

Distinct
(distinguishable)

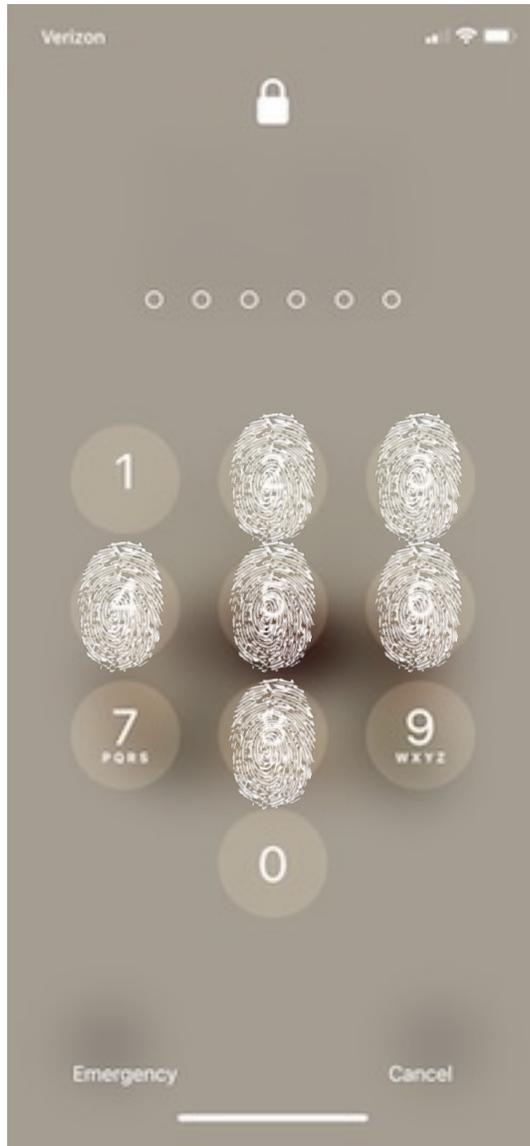
Some
distinct

$$n!$$

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

Unique 6-digit passcodes with **six** smudges

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

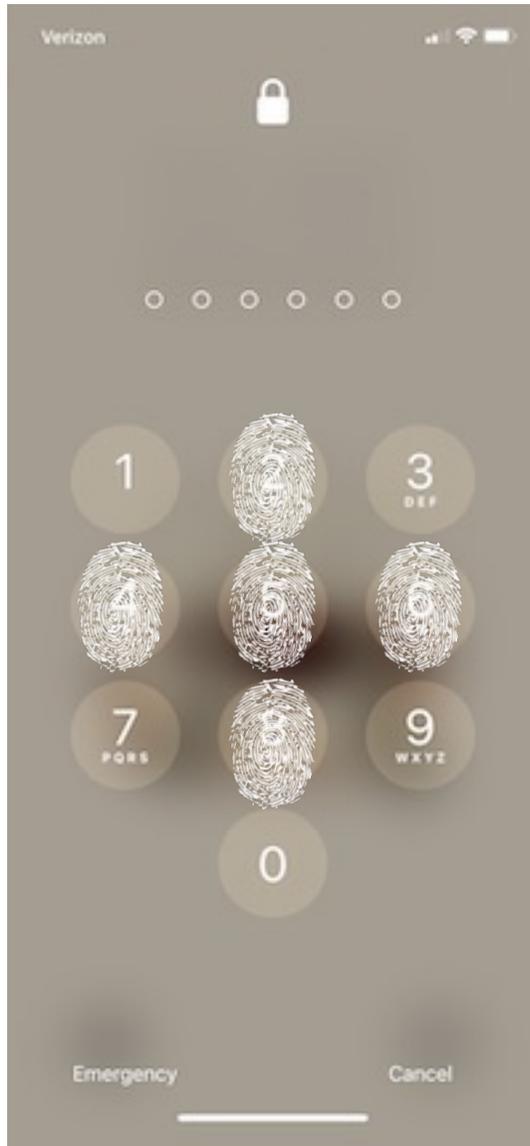


How many unique 6-digit passcodes are possible if a phone password uses each of **six** distinct numbers?

Total = $6!$
= 720 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)

$$\begin{aligned} \text{Total} &= 5 \times \frac{6!}{2!} \\ &= 1,800 \text{ passcodes} \end{aligned}$$

Combinations I

Summary of Combinatorics

Counting tasks on n objects

Sort objects
(permutations)

Choose k objects
(combinations)

Put objects in r
buckets

Distinct
(distinguishable)

Some
distinct

Distinct



$$n!$$

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

Combinations with cake

There are $n = 20$ people.

How many ways can we **choose** $k = 5$ people to get cake?

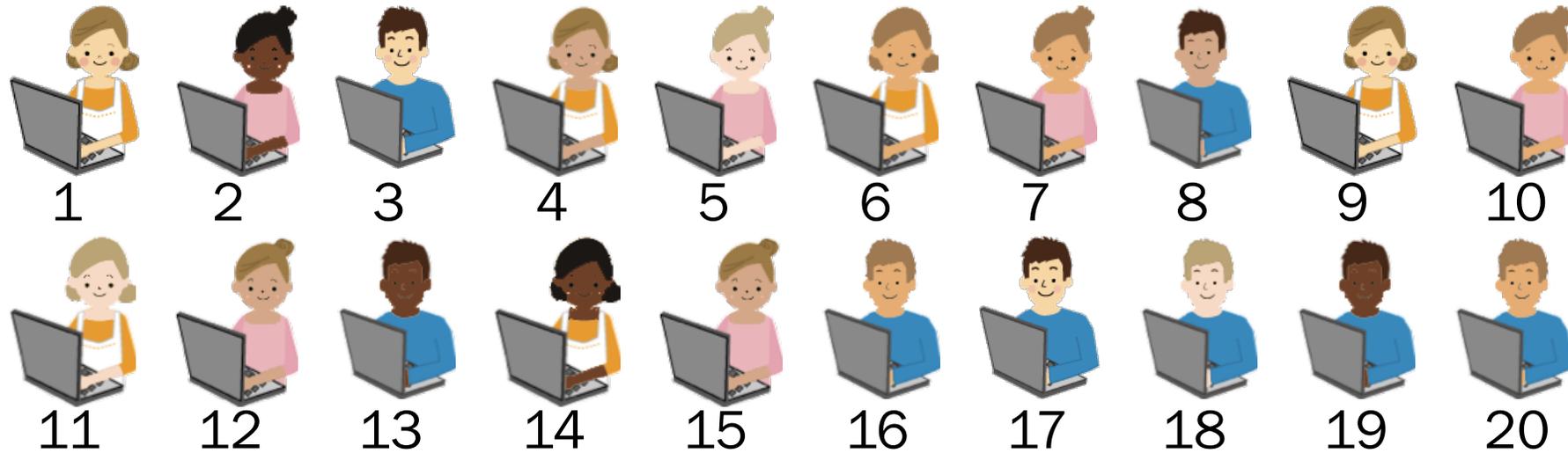


Consider the following generative process...

Combinations with cake

There are $n = 20$ people.

How many ways can we **choose** $k = 5$ people to get cake?



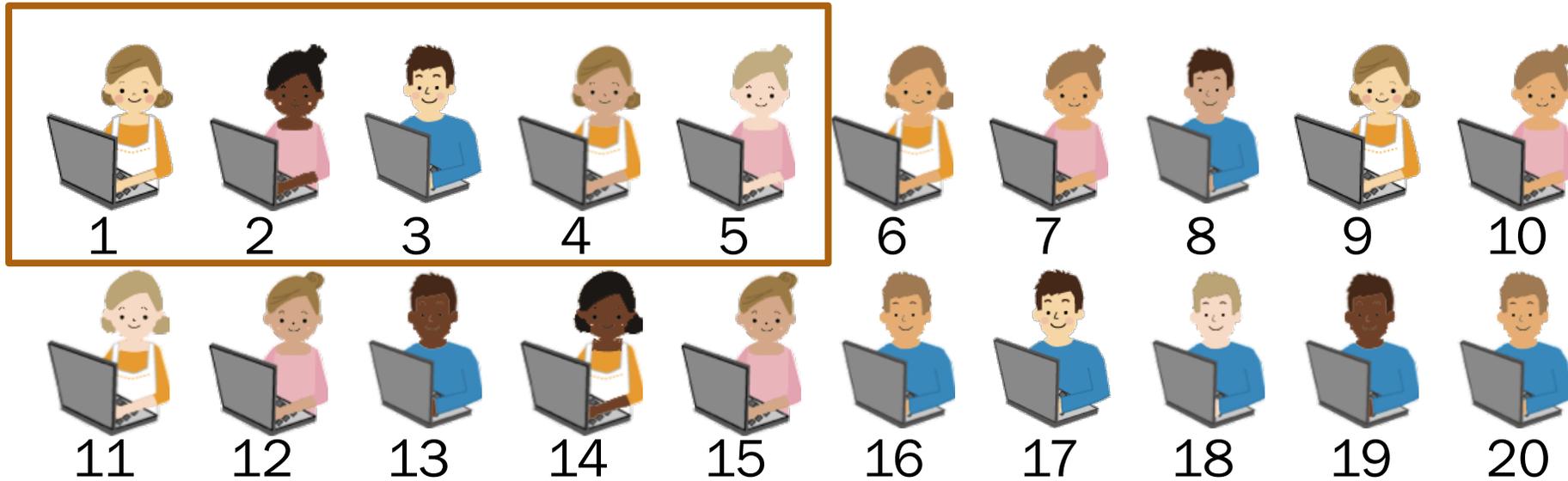
1. n people
get in line

$n!$ 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

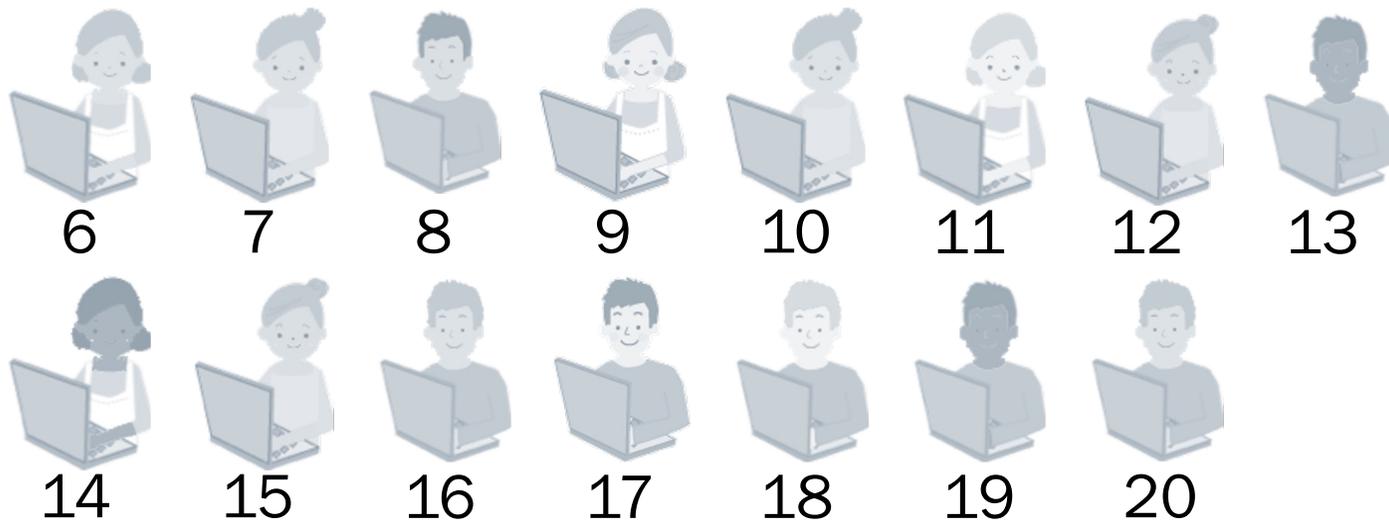
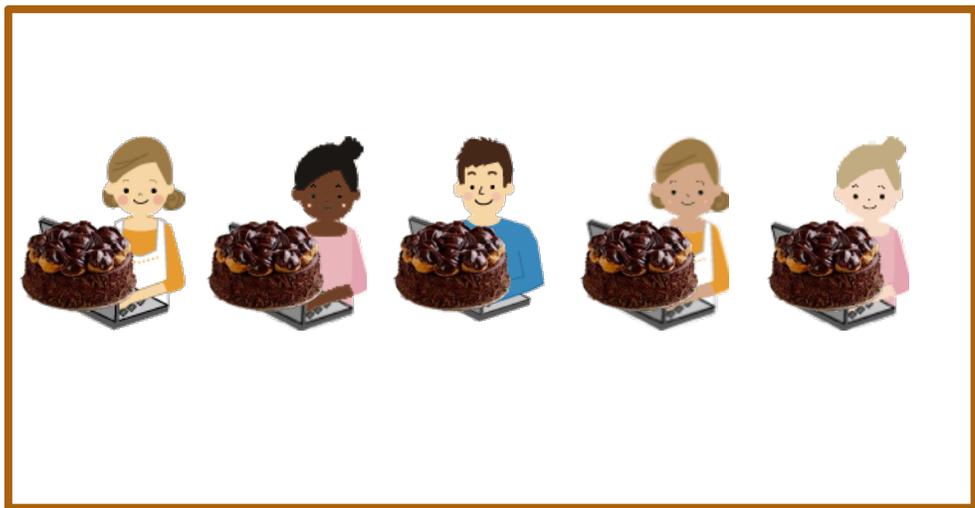
2. Put first k
in cake room

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

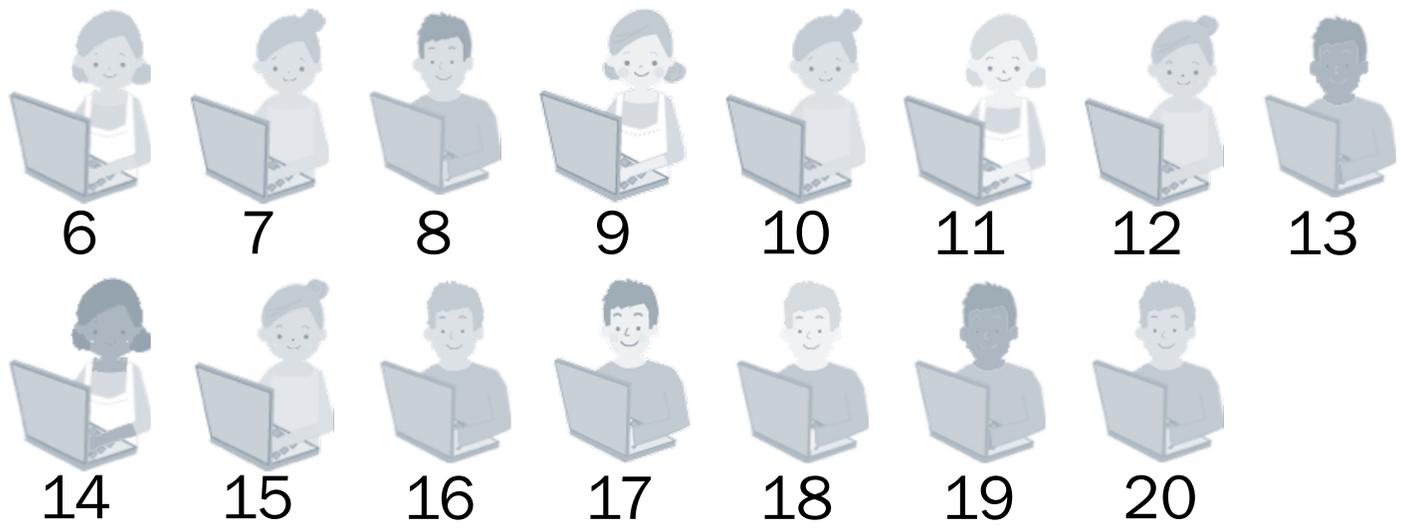
2. Put first k
in cake room

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

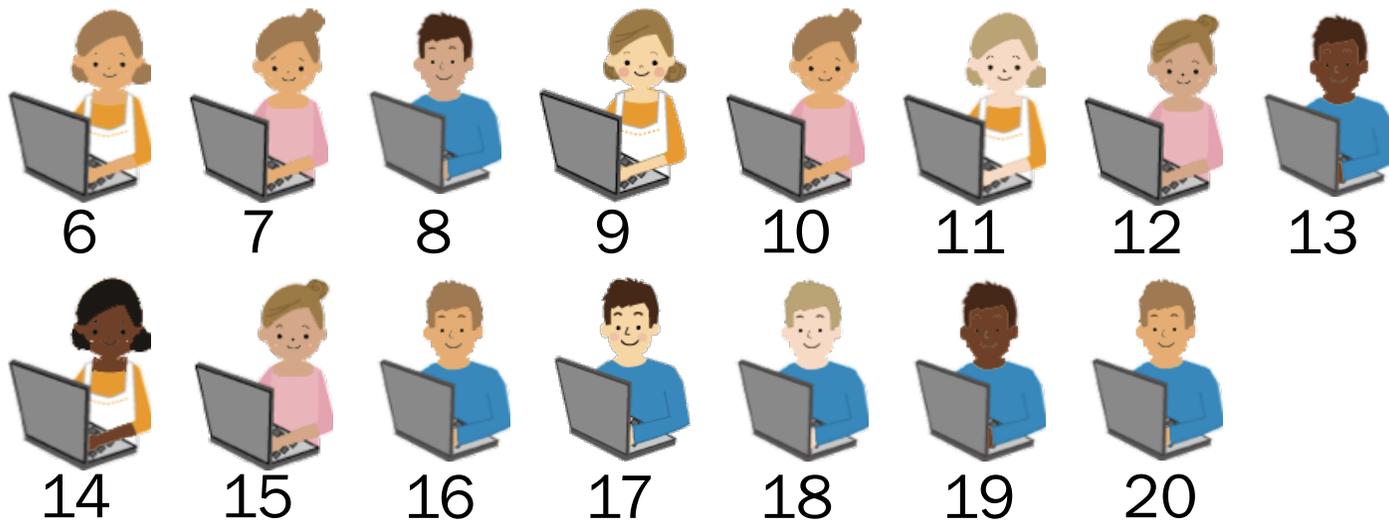
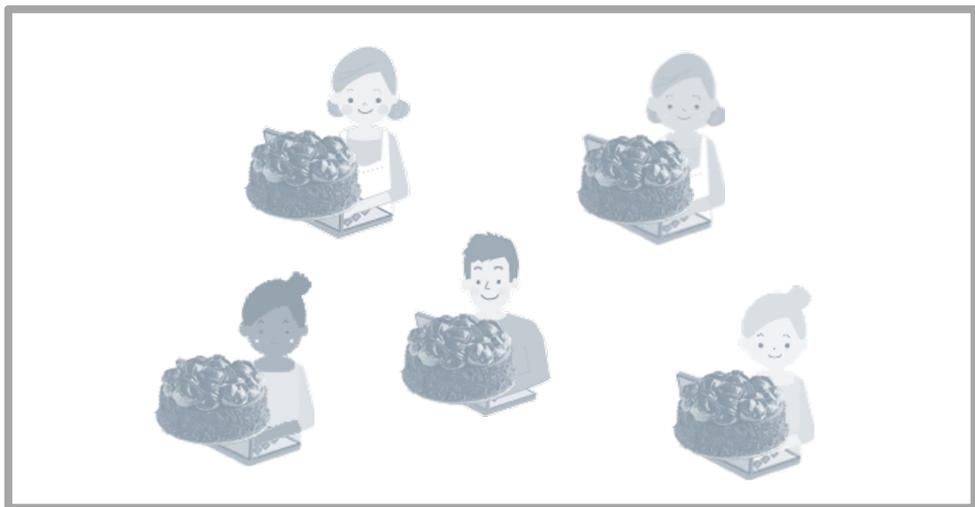
3. **Allow cake group to mingle**

$k!$ 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

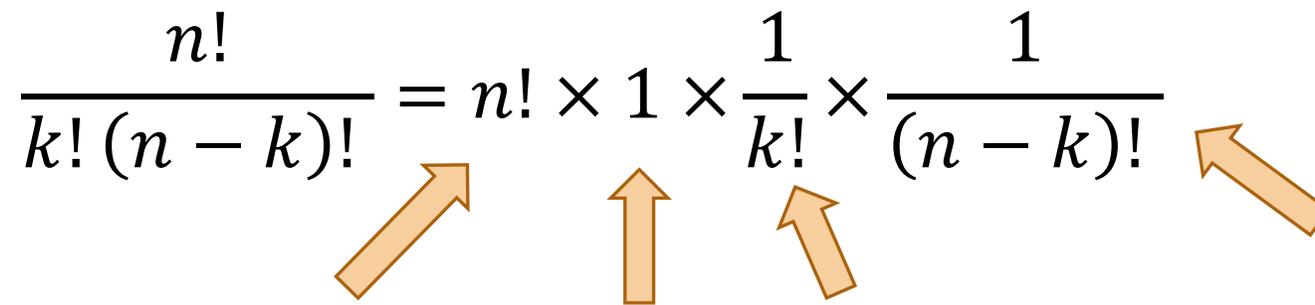
4. Allow non-cake
group to mingle

$(n - k)!$ 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

$$\frac{n!}{k! (n - k)!} = n! \times 1 \times \frac{1}{k!} \times \frac{1}{(n - k)!} = \binom{n}{k} \text{ Binomial coefficient}$$

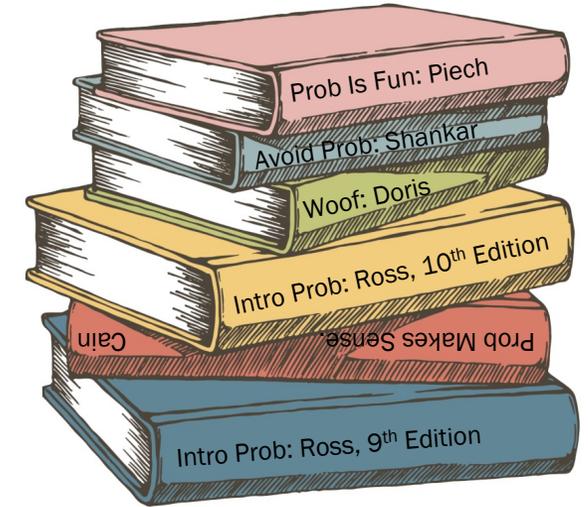
Fun Fact: $\binom{n}{k} = \binom{n}{n-k}$

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?



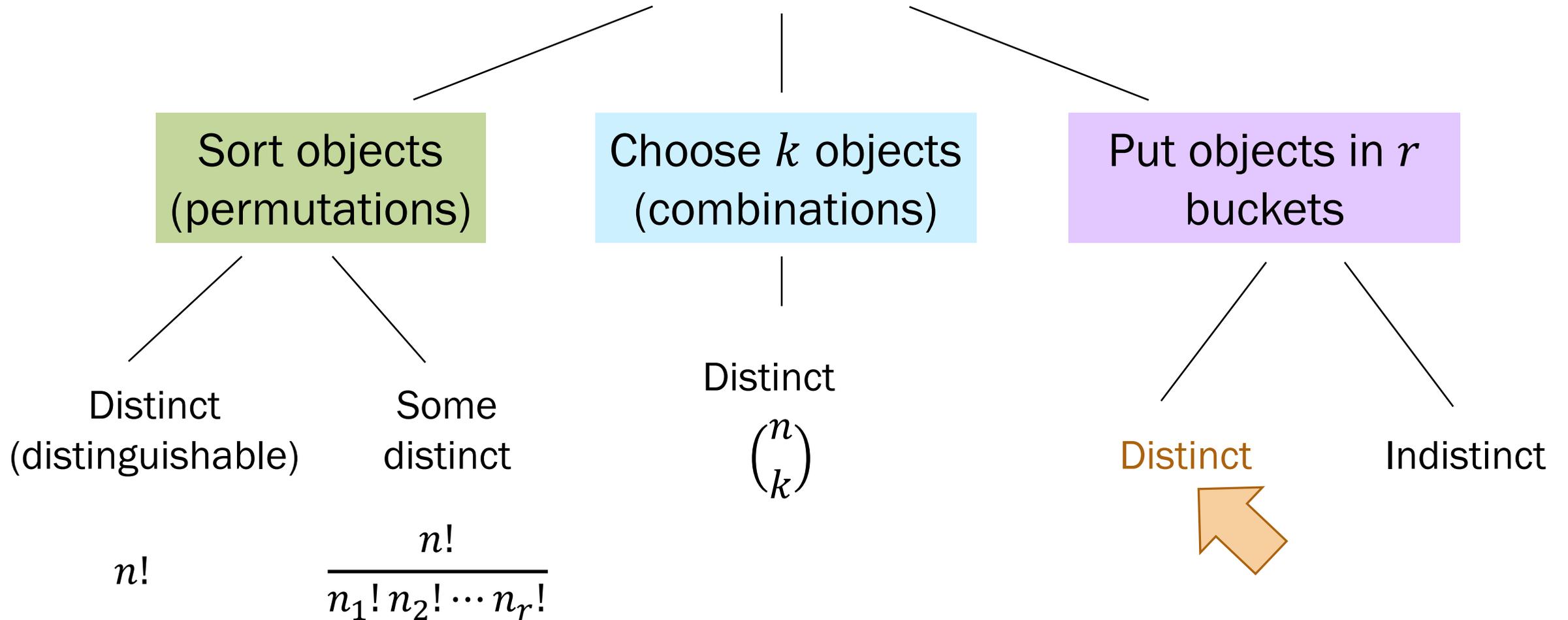
```
def main():  
    cards = make_deck()  
    all_hands = itertools.combinations(cards, 5)  
    for hand in all_hands:  
        print(hand)
```

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

Buckets and The Divider Method

Summary of Combinatorics

Counting tasks on n objects

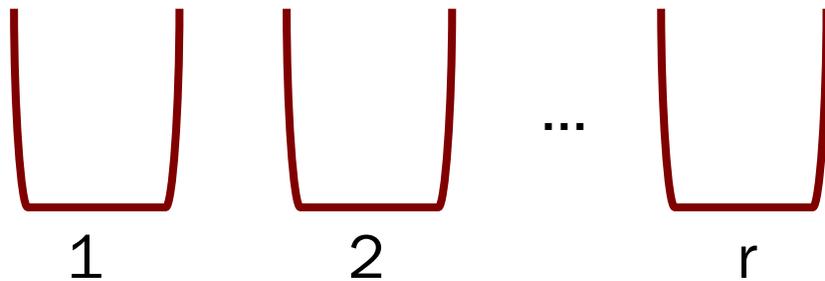
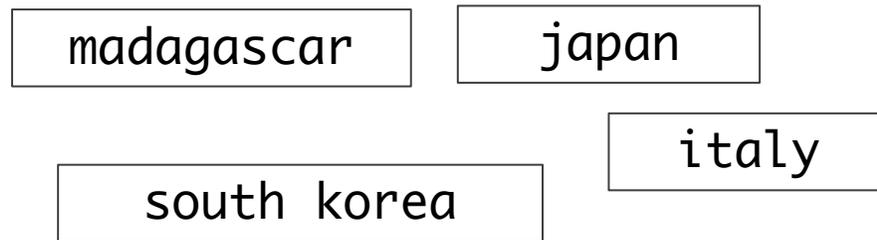


~~Balls and urns~~ Hash tables and **distinct** strings

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

Steps:

1. Bucket 1st string
2. Bucket 2nd string
- ...
- n . Bucket n^{th} string



r^n outcomes

Summary of Combinatorics

Counting tasks on n objects

Sort objects
(permutations)

Choose k objects
(combinations)

Put objects in r
buckets

Distinct
(distinguishable)

Some
distinct

Distinct

$$\binom{n}{k}$$

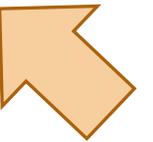
Distinct

Indistinct

$$n!$$

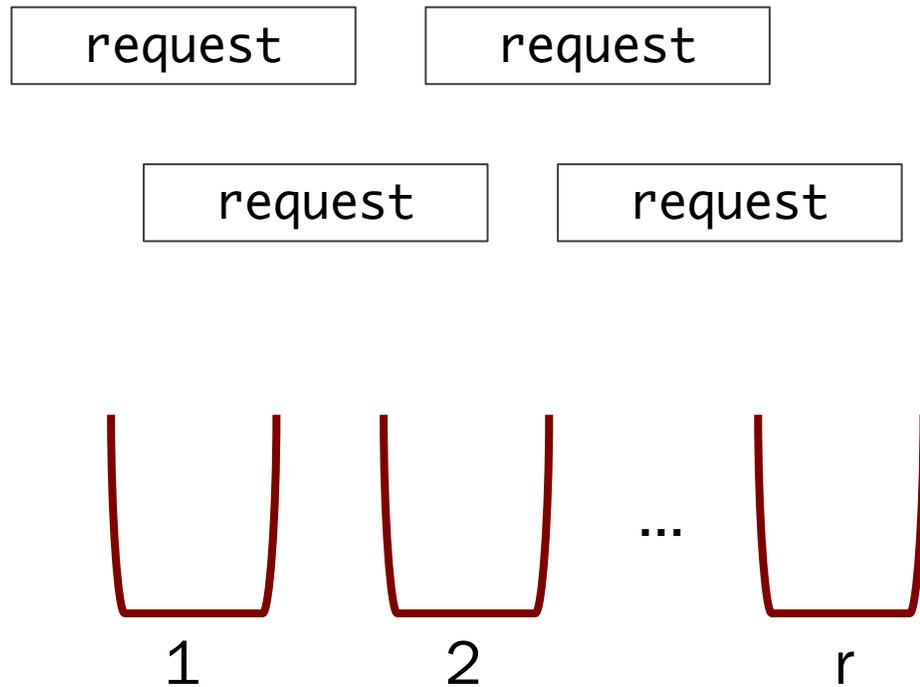
$$\frac{n!}{n_1! n_2! \cdots n_r!}$$

$$r^n$$



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,

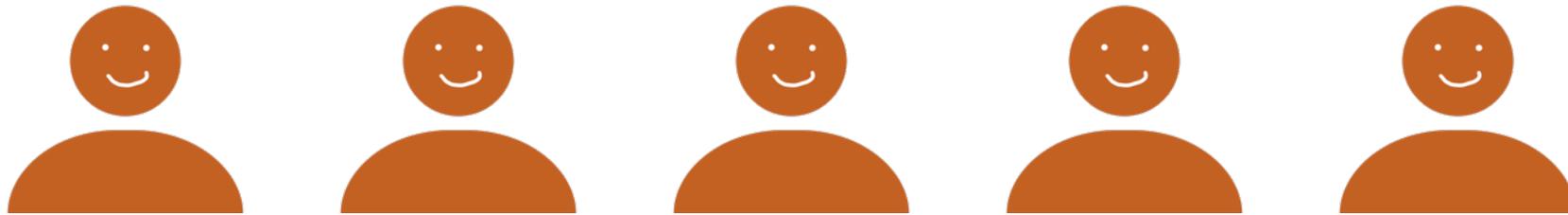
...

Server r has x_r requests

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

Bicycle helmet sales

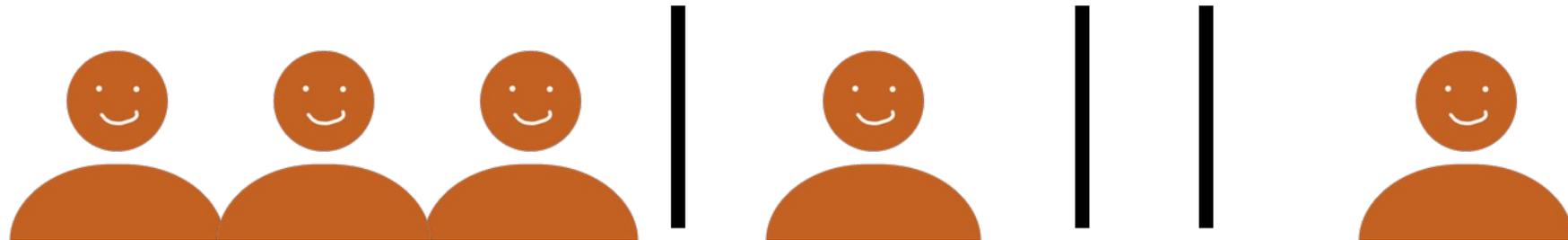
How many ways can we assign $n = 5$ indistinct children to $r = 4$ distinct bicycle helmet styles?



Bicycle helmet sales

1 possible assignment outcome:

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



Consider the following generative process...

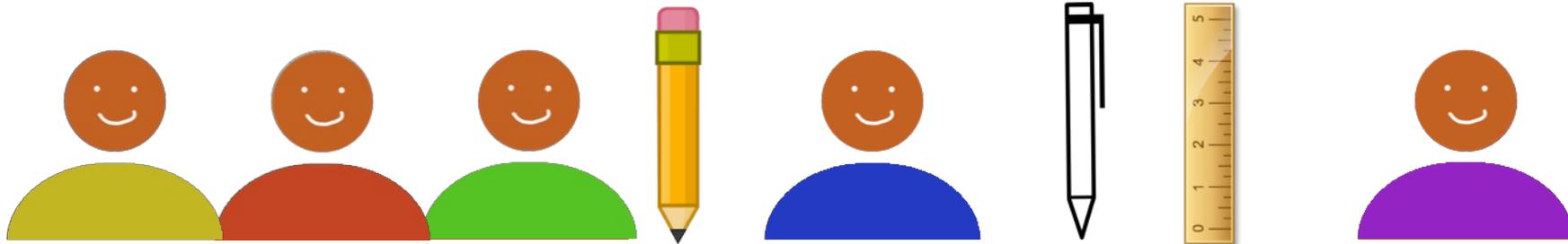


The divider method: A generative proof

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

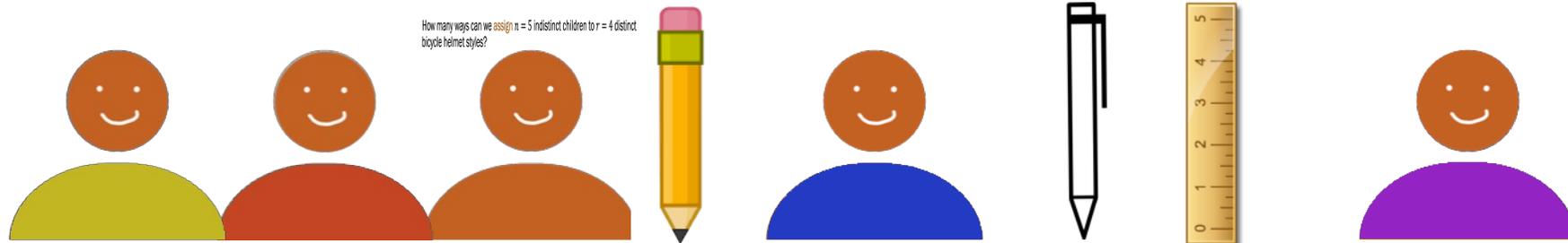


The divider method: A generative proof

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

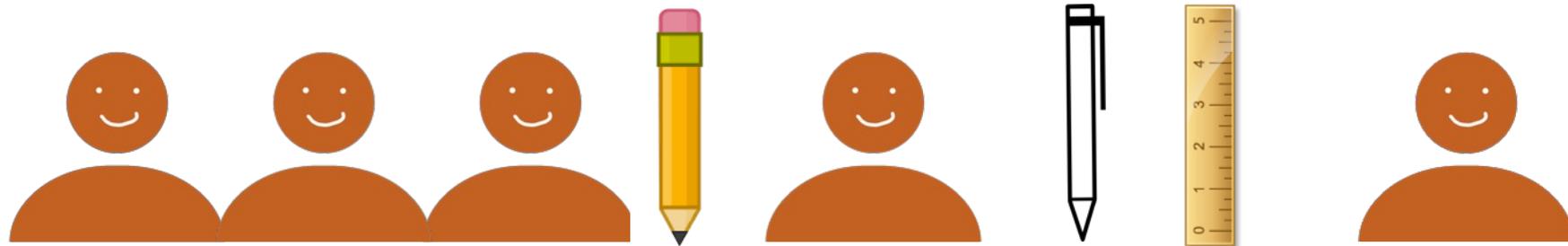
$$(n + r - 1)!$$

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

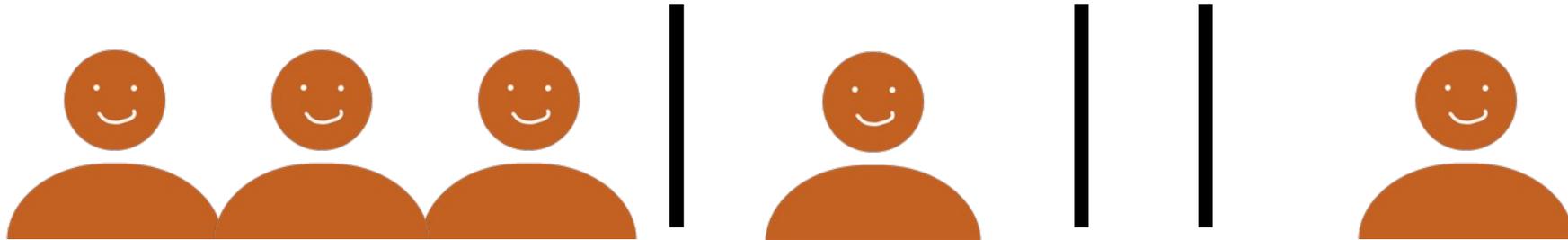
$$\frac{1}{n!}$$

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

How many integer solutions are there to the following equation:

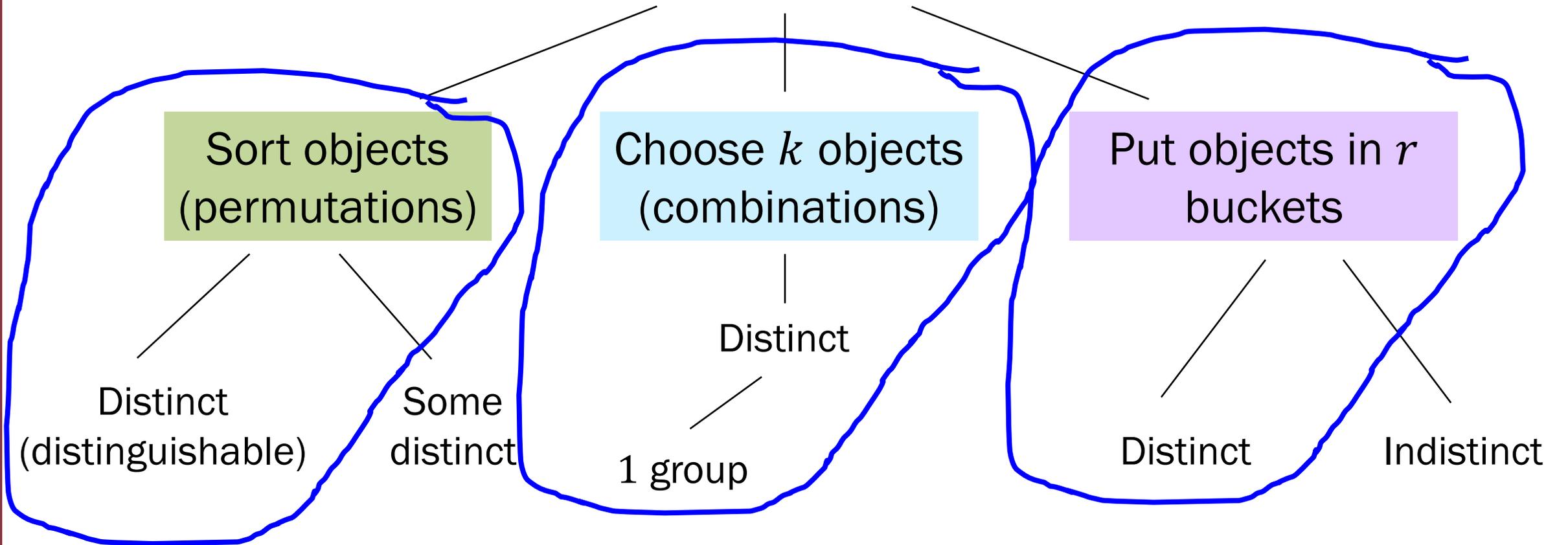
$$x_1 + x_2 + \cdots + x_r = n,$$

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

Positive integer equations can be solved with the divider method.

Summary of Combinatorics

Counting tasks on n objects

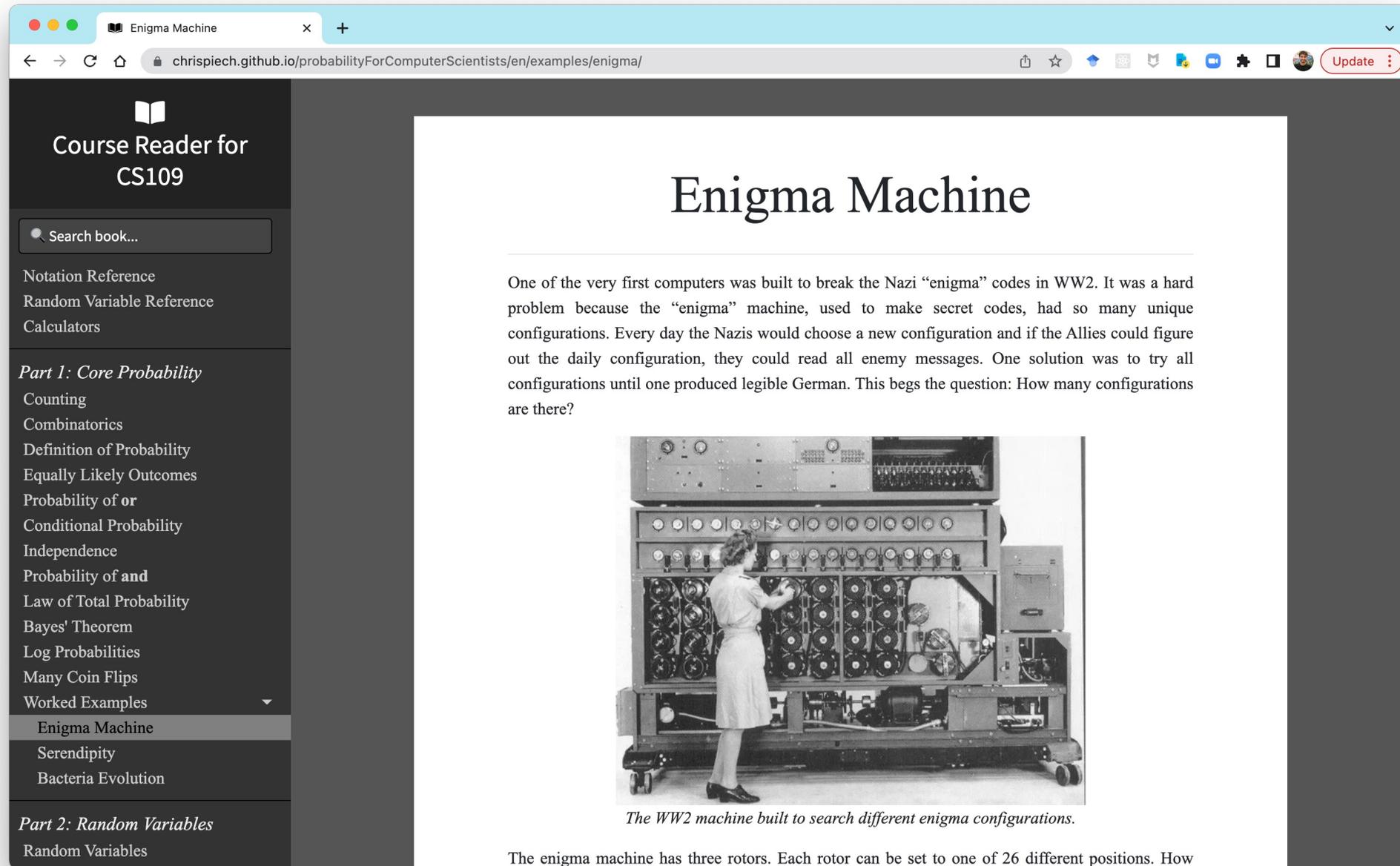


Ready for the first 4 problems (and the rest on Friday)

Now, more than ever, get started early

The screenshot shows a web browser window with the URL `cs109psets.netlify.app/win22/pset1/robot_paths`. The page title is "PS1 4. Robot Paths". On the left, a vertical navigation bar contains numbered buttons from 1 to 11. Buttons 1 through 10 are green with a checkmark, and button 4 is highlighted with a purple circle. The main content area contains the text: "Imagine you have a robot (🤖) that lives on an 7 x 8 grid (it has 7 rows and 8 columns). The robot starts in cell (1, 1) and can take steps either to the right or down (no left or up steps). How many distinct paths can the robot take to the destination (▶) in cell (7, 8)?" Below the text is a 7x8 grid. A robot icon is at the top-left cell (1,1) and a red flag icon is at the bottom-right cell (7,8). The grid is labeled "7 rows" on the left and "8 columns" at the bottom. To the right of the grid is an "Answer Editor" section with a "Numeric Answer:" field containing the text "Enter your answer" and a "Check Answer" button. Below that is an "Explanation:" section with a rich text editor toolbar (Block LaTeX, Image, Bold, Italic, Underline) and a large empty text area. At the bottom of the page are "Previous Question" and "Next Question" buttons.

Want to go deep?



The screenshot shows a web browser window with the address bar displaying `chrispiech.github.io/probabilityForComputerScientists/en/examples/enigma/`. The page content includes a sidebar on the left with a search bar and a list of topics. The main content area features a title "Enigma Machine" and a paragraph of text. Below the text is a black and white photograph of a woman operating a large, complex machine with many rotors. The sidebar on the left is titled "Course Reader for CS109" and contains a search bar and a list of topics. The "Enigma Machine" topic is highlighted in the "Worked Examples" section. The main content area has a title "Enigma Machine" and a paragraph of text. Below the text is a black and white photograph of a woman operating a large, complex machine with many rotors. The caption below the photo reads: "The WW2 machine built to search different enigma configurations." The footer of the page includes the text "The enigma machine has three rotors. Each rotor can be set to one of 26 different positions. How" and the Stanford University logo.

Course Reader for CS109

Search book...

Notation Reference
Random Variable Reference
Calculators

Part 1: Core Probability

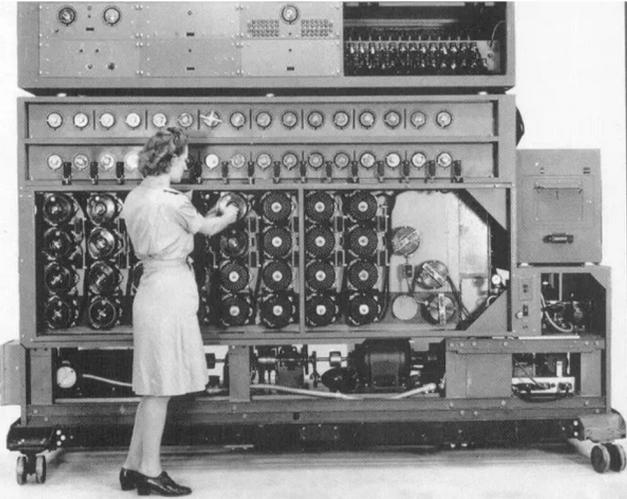
- Counting
- Combinatorics
- Definition of Probability
- Equally Likely Outcomes
- Probability of **or**
- Conditional Probability
- Independence
- Probability of **and**
- Law of Total Probability
- Bayes' Theorem
- Log Probabilities
- Many Coin Flips
- Worked Examples
 - Enigma Machine**
 - Serendipity
 - Bacteria Evolution

Part 2: Random Variables

- Random Variables

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?