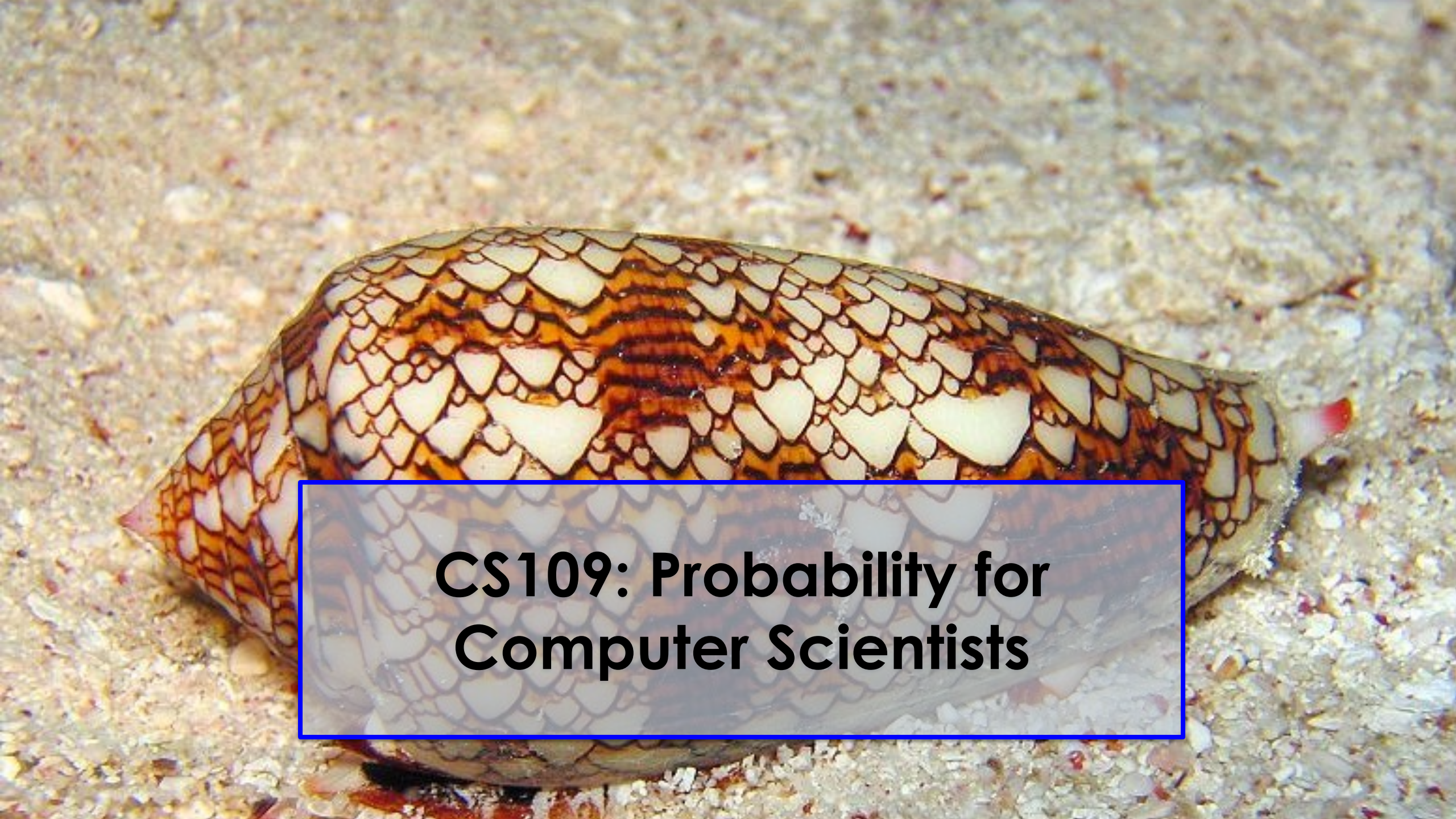
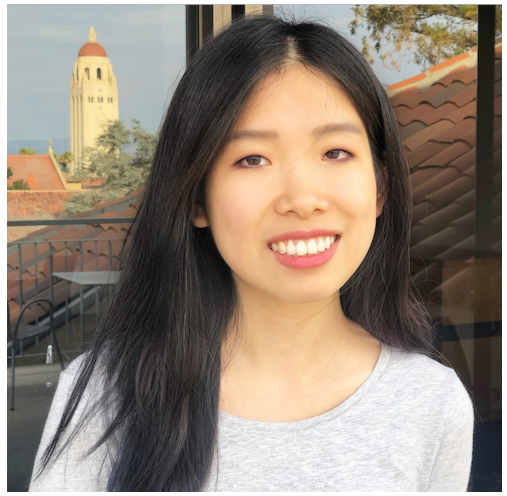


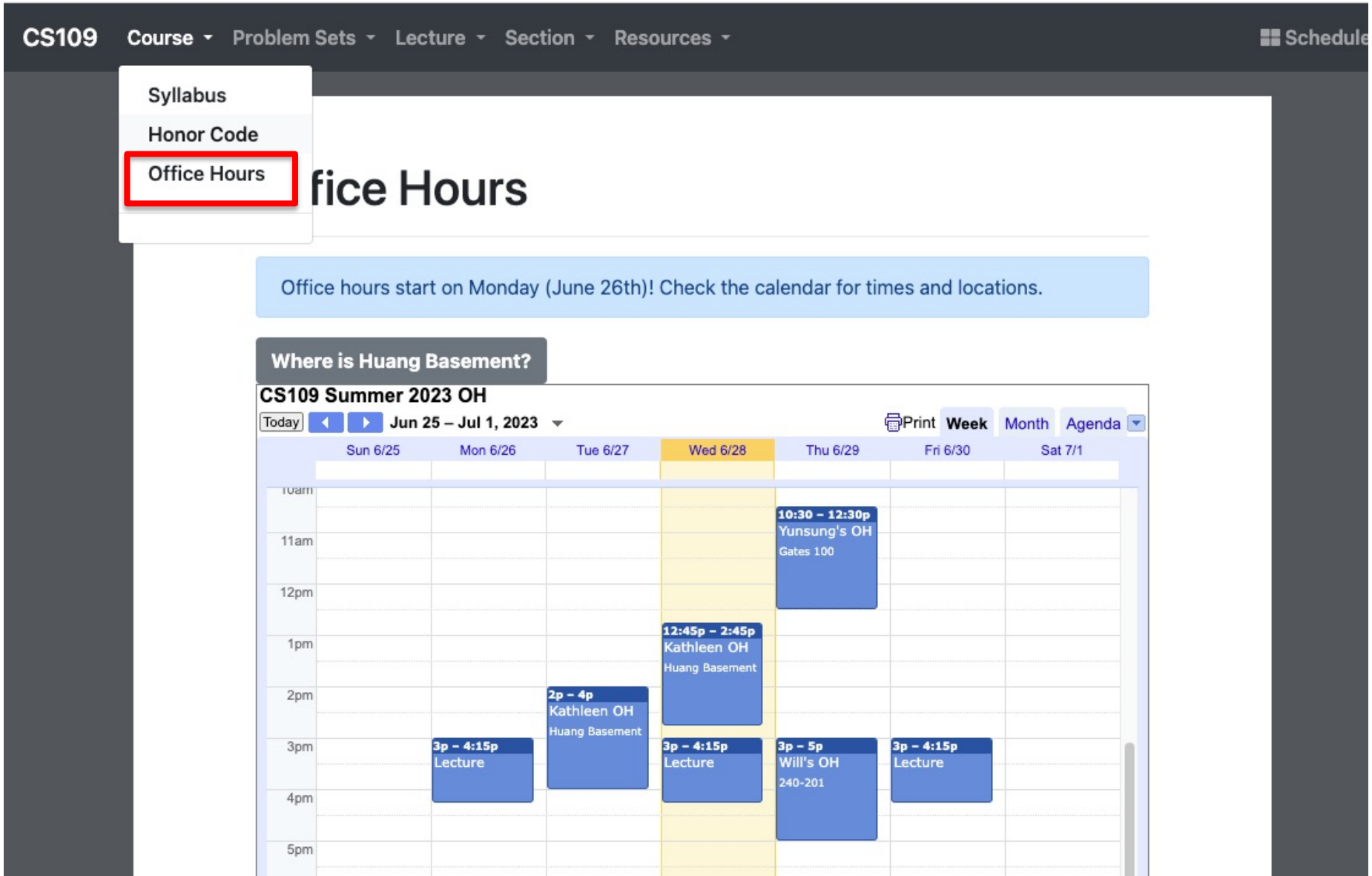
CS109: Probability for Computer Scientists

New Teaching Staff Member!!



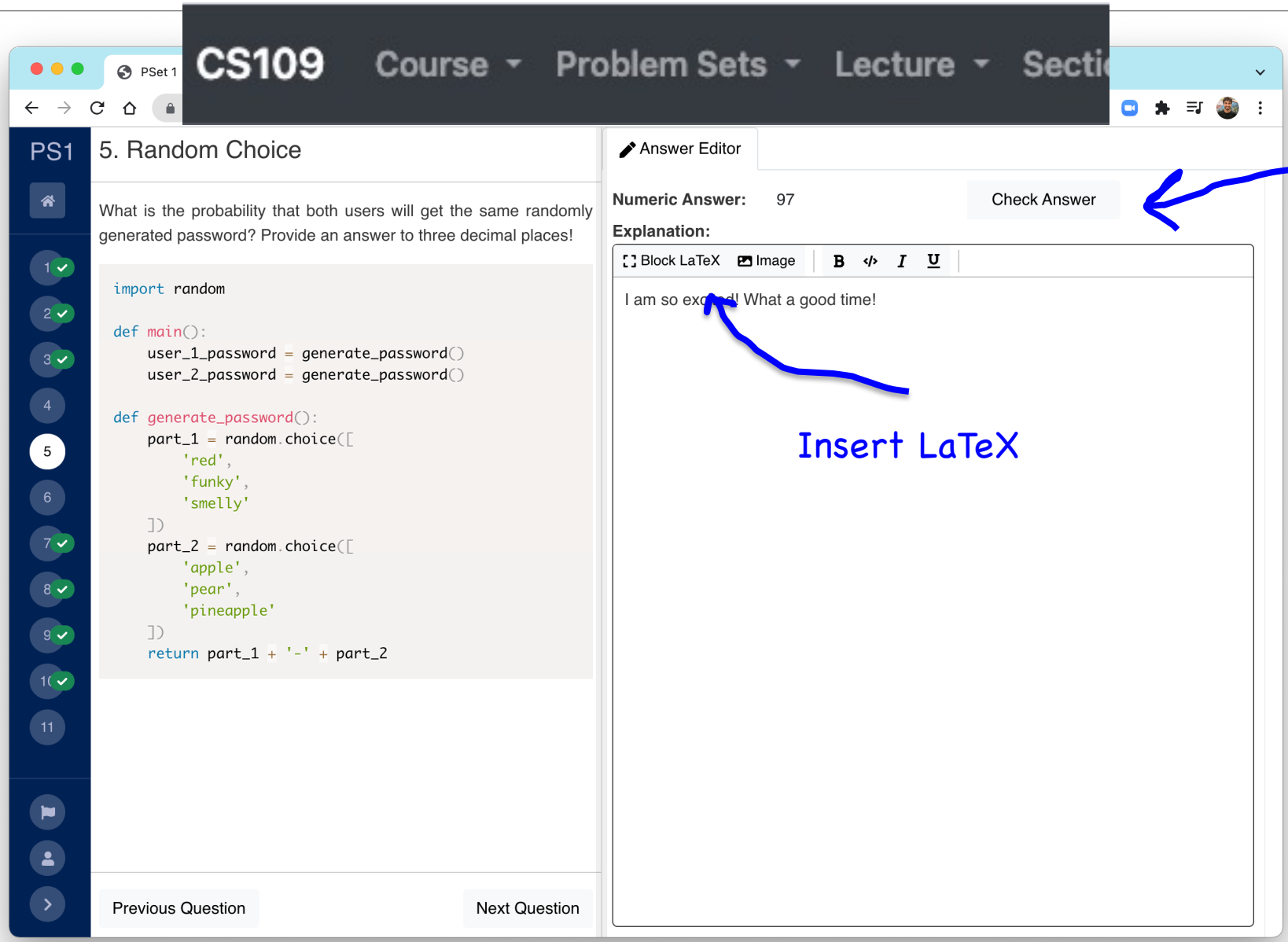
Tori Qiu

Office Hours

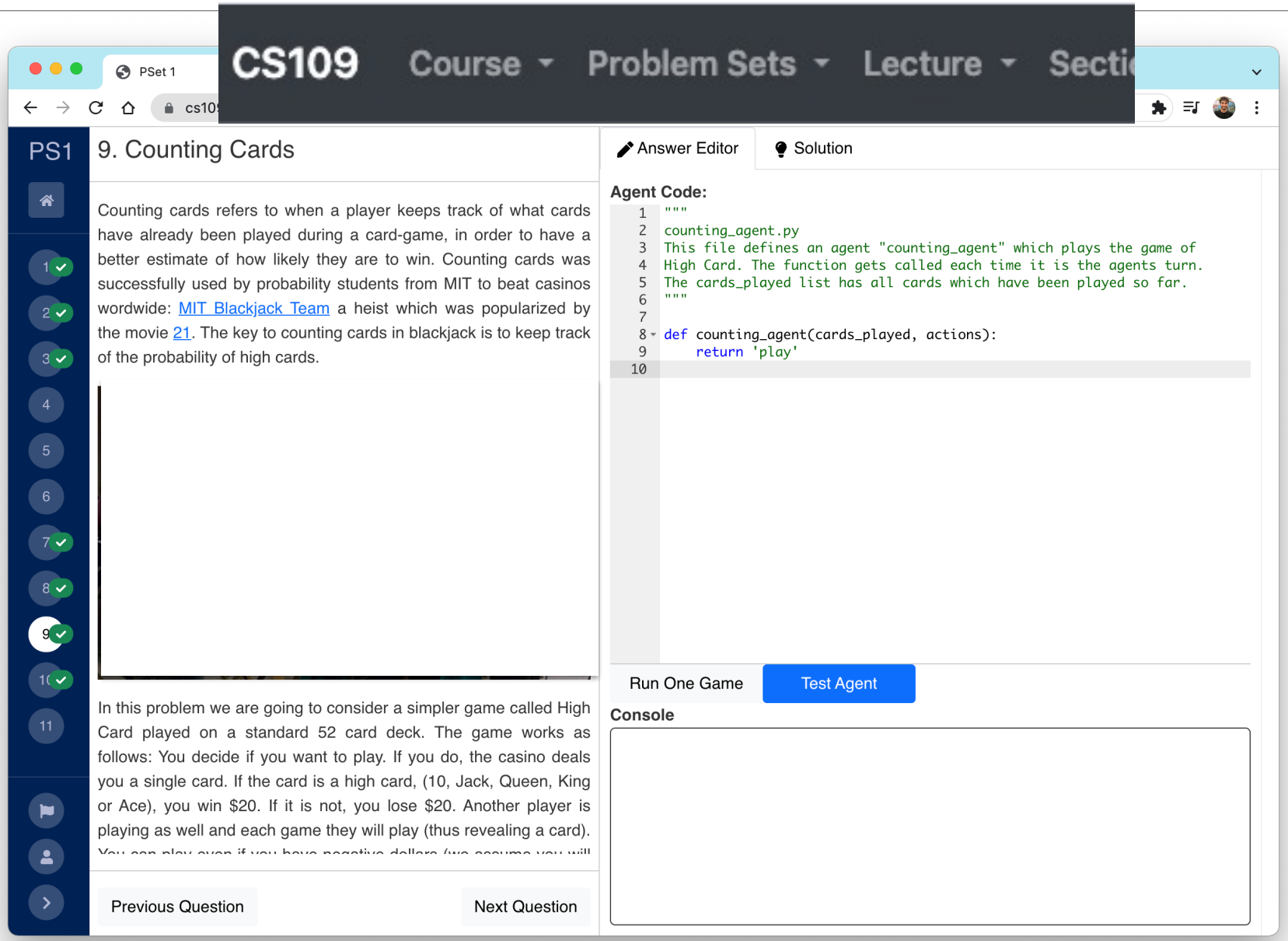


Section Signups Tomorrow

Problem Set #1 is out



Write an Agent



Python Review Session

CS109 Course Problem Sets Lecture Section Resources

Course Reader
Python Review
Latex Cheat Sheet

Python for Proba

We'll hold two Python review sessions throughout the quarter to get you up to speed on what you'll need for the problem sets.

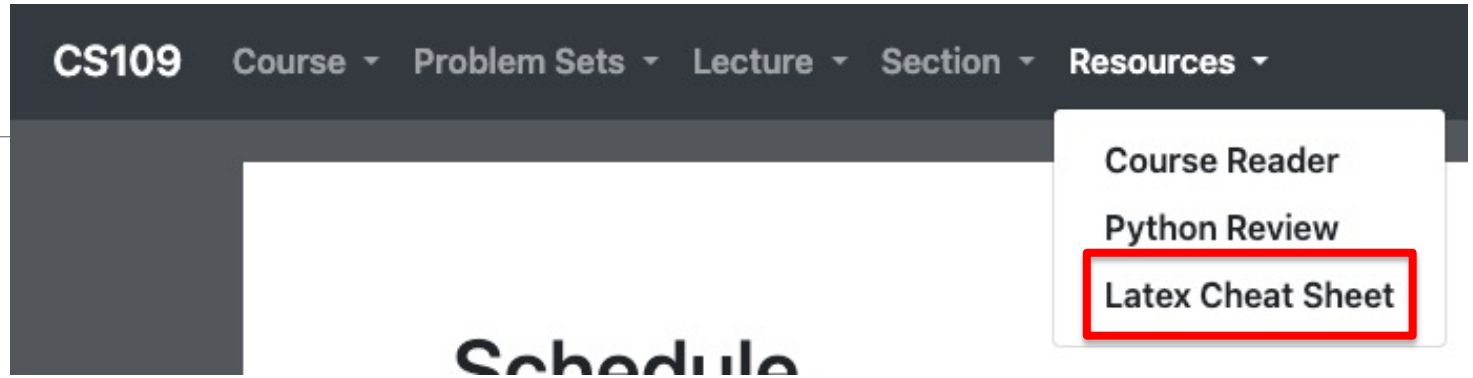
If you want to get more python practice, you can also check out **Python tutorial notebook** (make sure you are logged in with your Stanford accout)!

- **Session #1**: Thursday June 29th, 4pm Pacific Time. Intro, running programs, Python basics.
  Zoom:
  Link
- **Session #2**: Thursday July 6th, 4:30pm Pacific Time. NumPy arrays and random numbers.
  Zoom:
  Link

This and Next Thursday

Online and will be recorded!

Learn LaTex



We have 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}
```

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

Done

Honor Code

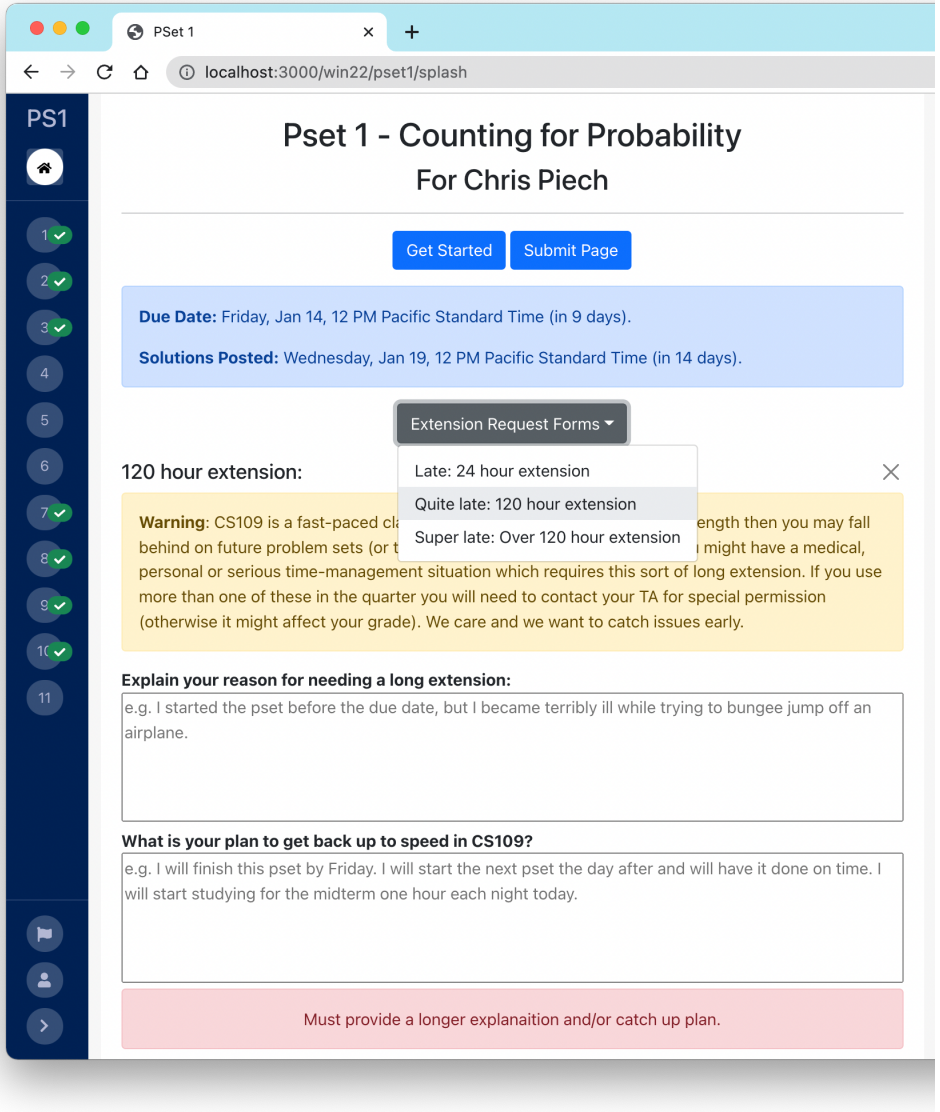
Always remember: You need to be able to recreate your ability on an exam. And in the real world. This is a foundation course. **Cheating in CS109 is cheating yourself**.

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:

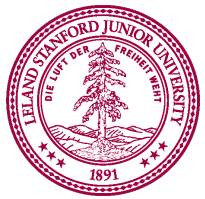
A. You need more than 2 long extensions / qtr
B. Extension past the hard deadline

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 cs109@cs.stanford.edu or email any course staff individually. We 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 :-)

How to Get the Most Out of Lectures?

Everyone has different styles of learning!

A lot of the technical details appear in the course reader.

It might help to follow along with the big ideas during class, and refer back to course reader/lectures if you want to fill in details.

BUT!!

Find out what works best for **you** ☺ !!

Review

CS109: From Counting to 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

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

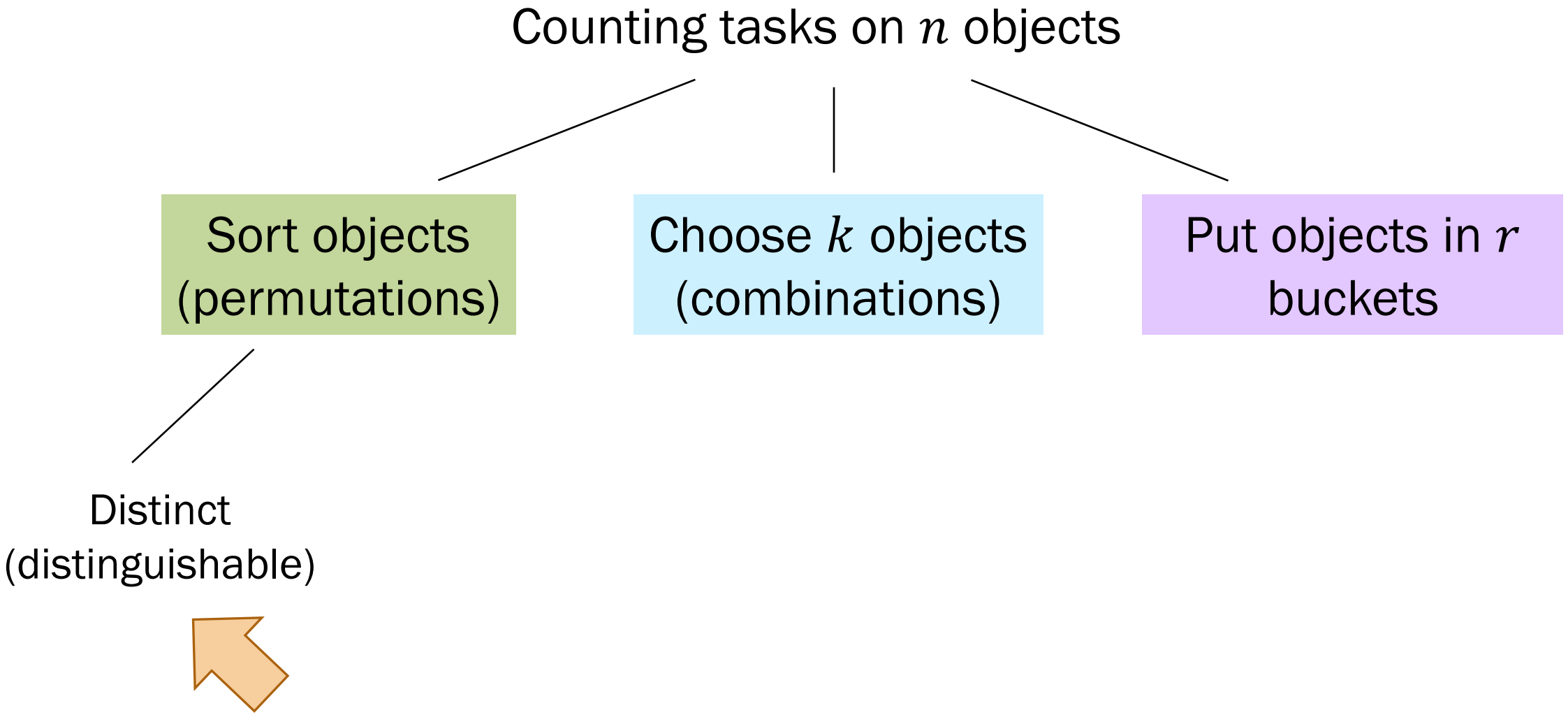
1. Strings
   - How many *different* orderings of letters are possible for the string BOBA?

BOBA, ABOB, OBBA…

End Review

Permutations I

Summary of Combinatorics



Orderings of Letters

How many letter orderings are possible for the following strings?

1. GATES

Orderings of Letters

gates	getsa	atges	asteg	tegas	egtsa	esgat	sateg
gatse	gesat	atgse	asegt	tegsa	egsat	esgta	saegt
gaets	gesta	ategs	asetg	teags	egsta	esagt	saetg
gaest	gsate	atesg	tgaes	teasg	eagts	esatg	stgae
gaste	gsaet	atsge	tgase	tesga	eagst	estga	stgea
gaset	gstae	atseg	tgeas	tesag	eatgs	estag	stage
gtaes	gstea	aegts	tgesa	tsgae	eatsg	sgate	staeg
gtase	gseat	aegst	tgsae	tsgea	easgt	sgaet	stega
gteas	gseta	aetgs	tgsea	tsage	eastg	sgtae	steag
gtesa	agtes	aetsg	tages	tsaeg	etgas	sgtea	segat
gtsae	agtse	aesgt	tagse	tsega	etgsa	sgeat	segta
gtsea	agets	aestg	taegs	tseag	etags	sgeta	seagt
geats	agest	asgte	taesg	egats	etasg	sagte	seatg
geast	agste	asget	tasge	egast	etsga	saget	setga
getas	agset	astge	taseg	egtas	etsag	satge	setag

Orderings of letters

Step 1:
Chose first letter

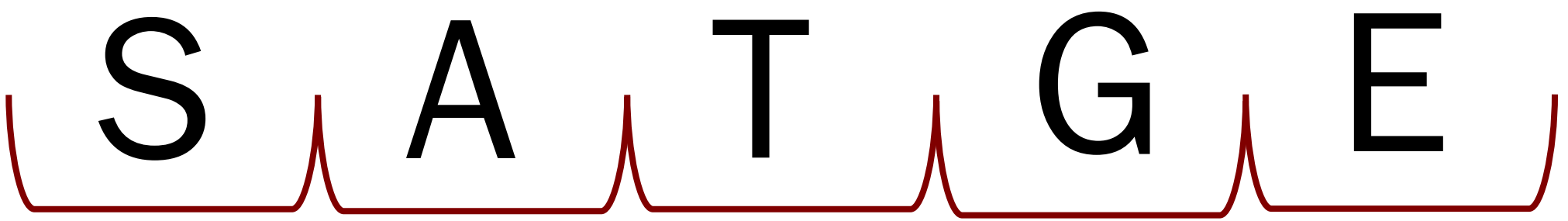
Step 2:
Chose 2nd letter

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)

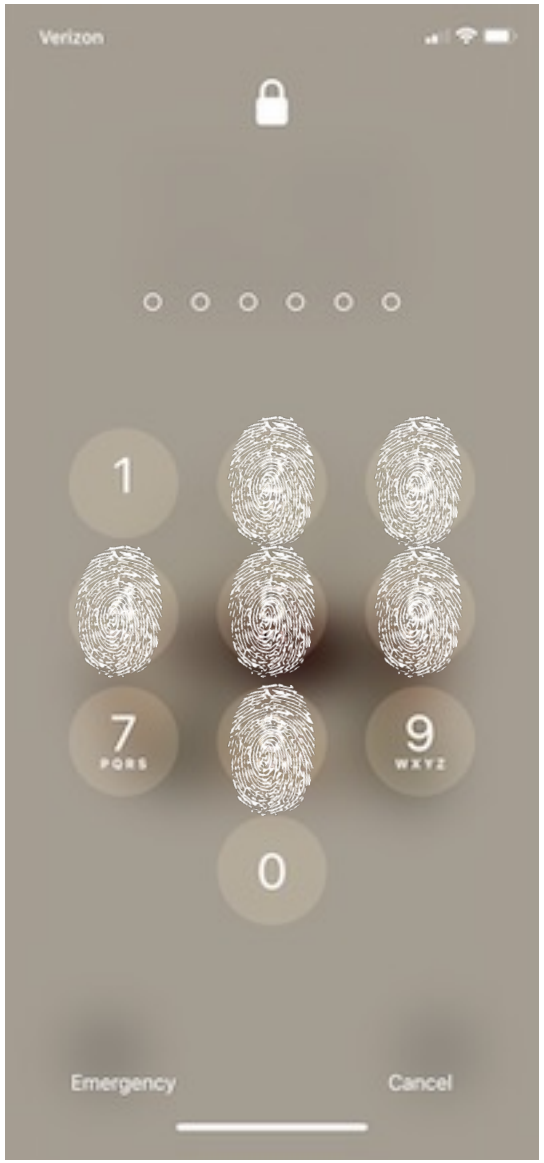
Permutations

A permutation is an ordered arrangement of objects.

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

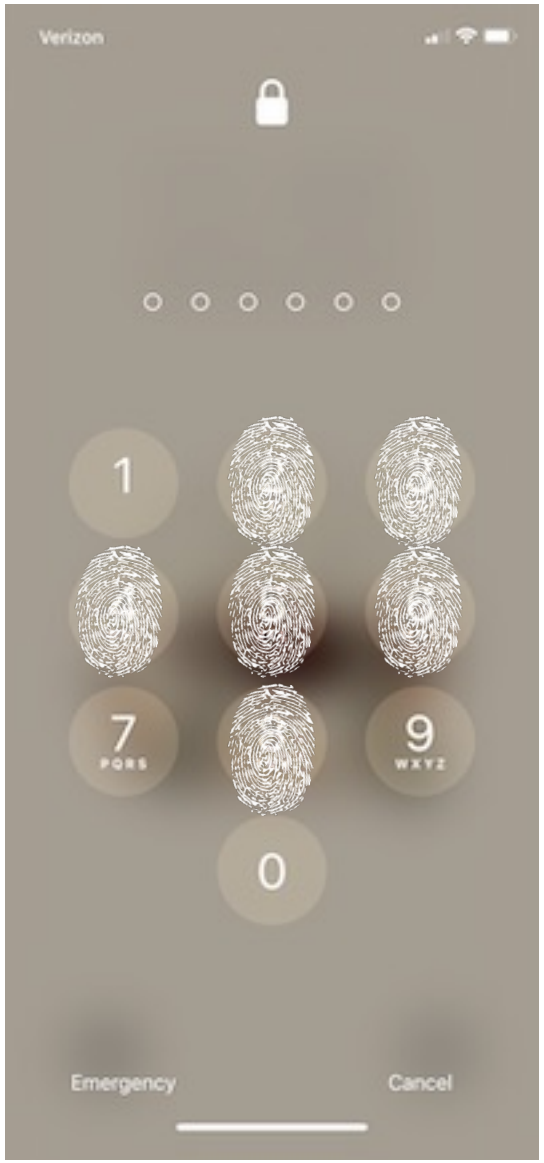
$$n! = n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1.$$

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 unique 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 indistinct objects?

Sort n distinct objects

Sort n distinct objects

1st 2nd 3rd 4th 5th

Steps:

1. Choose 1st can — 5 options
2. Choose 2nd can — 4 options

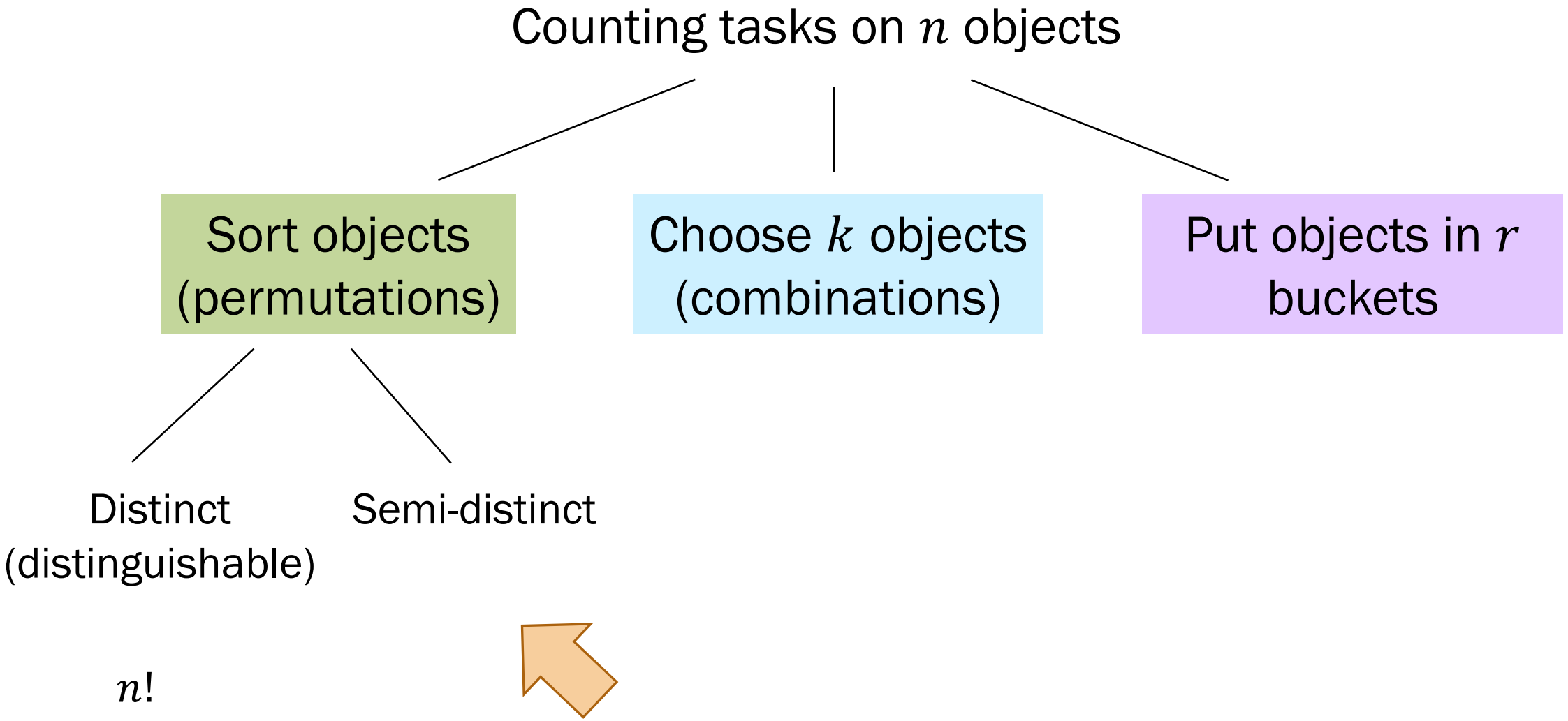
…

5. Choose 5th can — 1 option

$$\text{Total} = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

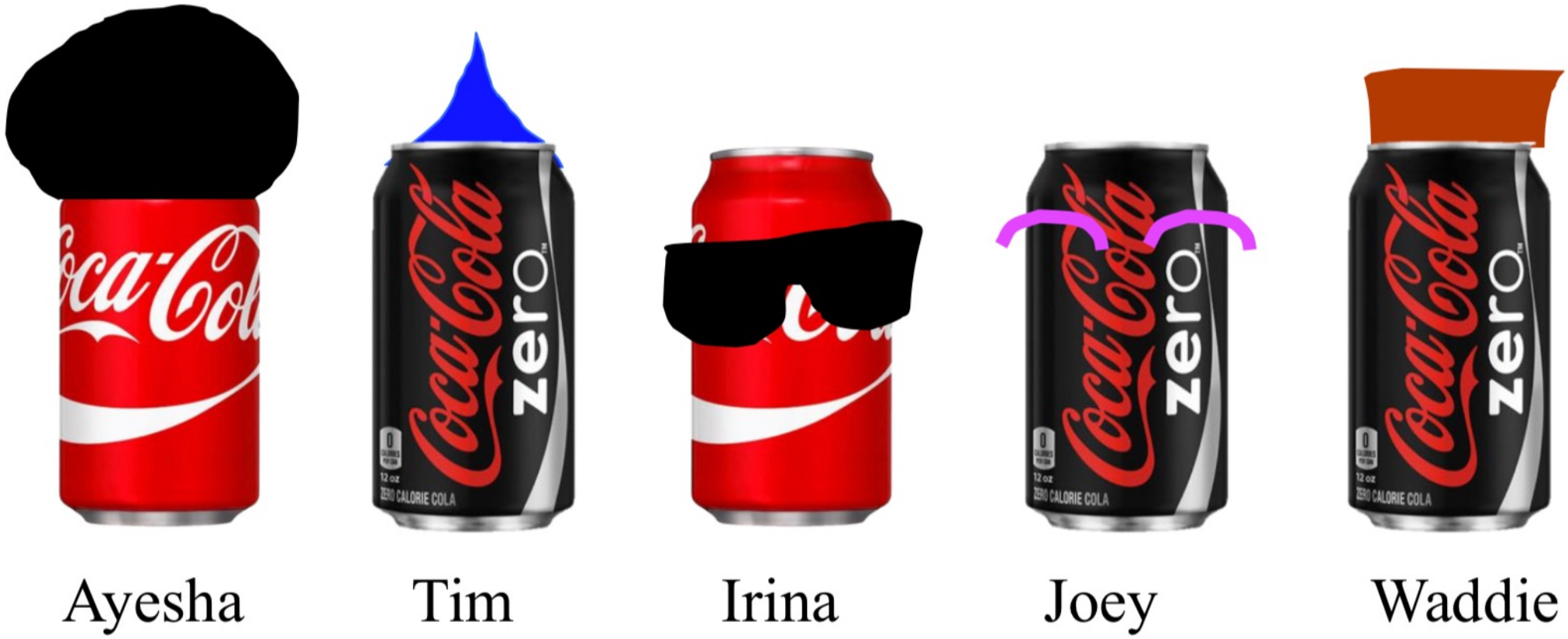
Permutations II

Summary of Combinatorics



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

Ayesha Tim Irina Joey Waddie

Some indistinct

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 × 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:

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

In combinatorics it often helps to overcount permutations and fix for the overcounting

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

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

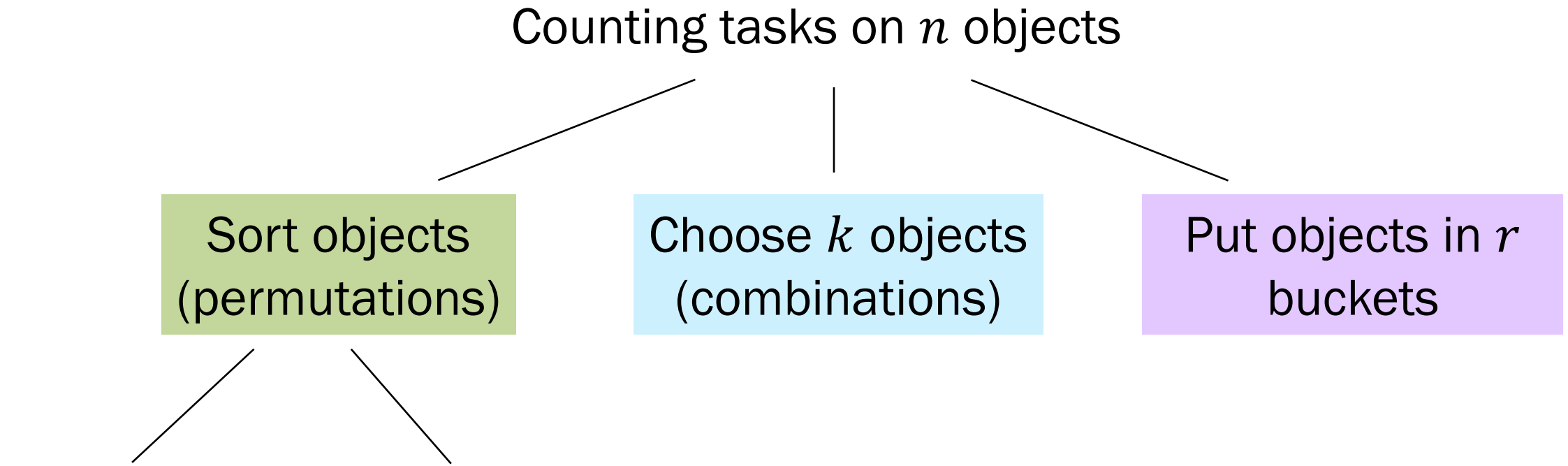
To the Code!

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

```python
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)
```

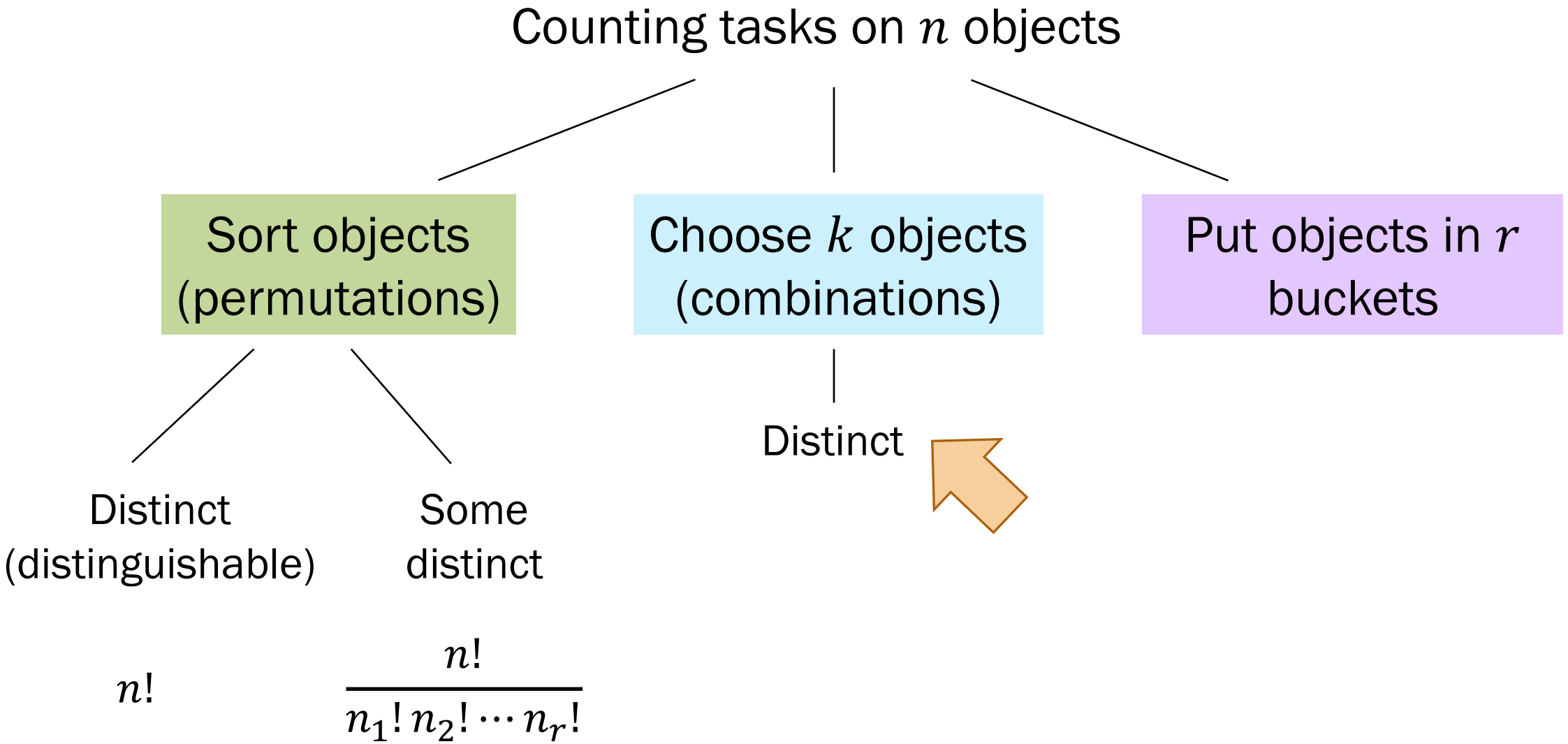
```python
import math

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

Summary of Combinatorics



Combinations I

Summary of Combinatorics



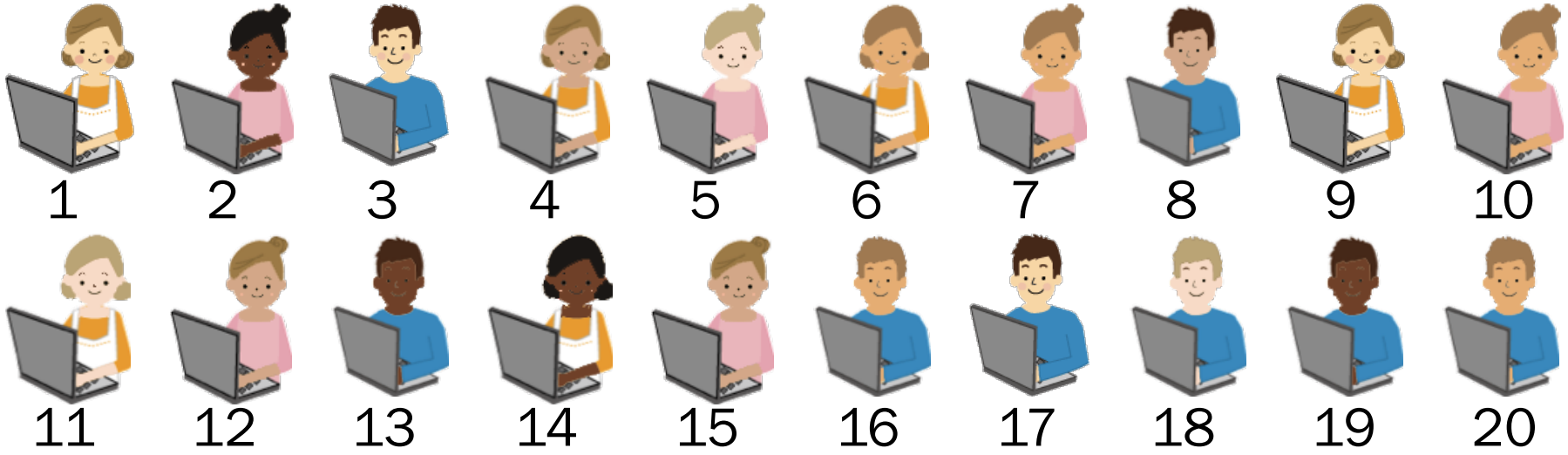
Combinations with cake

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?

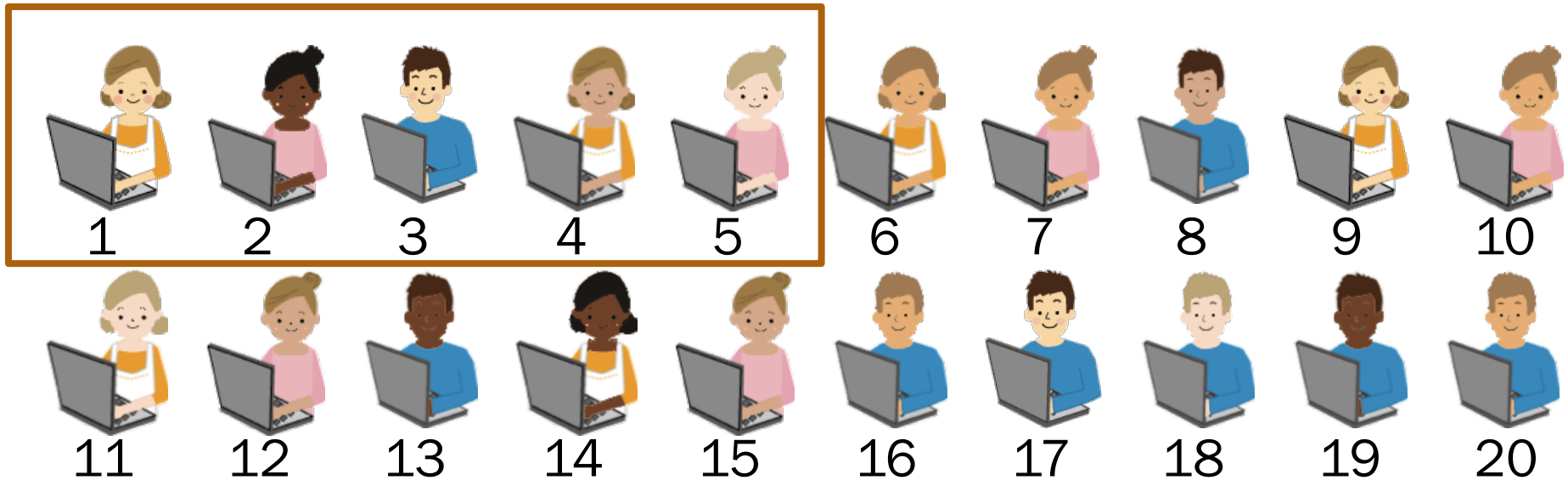


1. n people get in line

   $n!$ 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

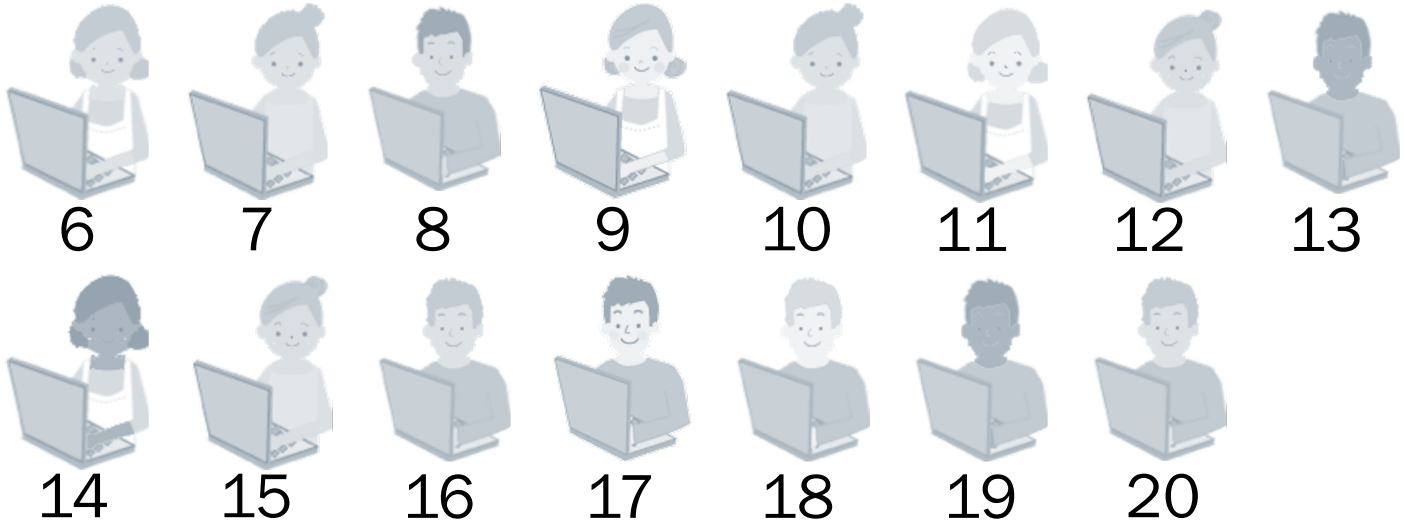
   $n!$ ways

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

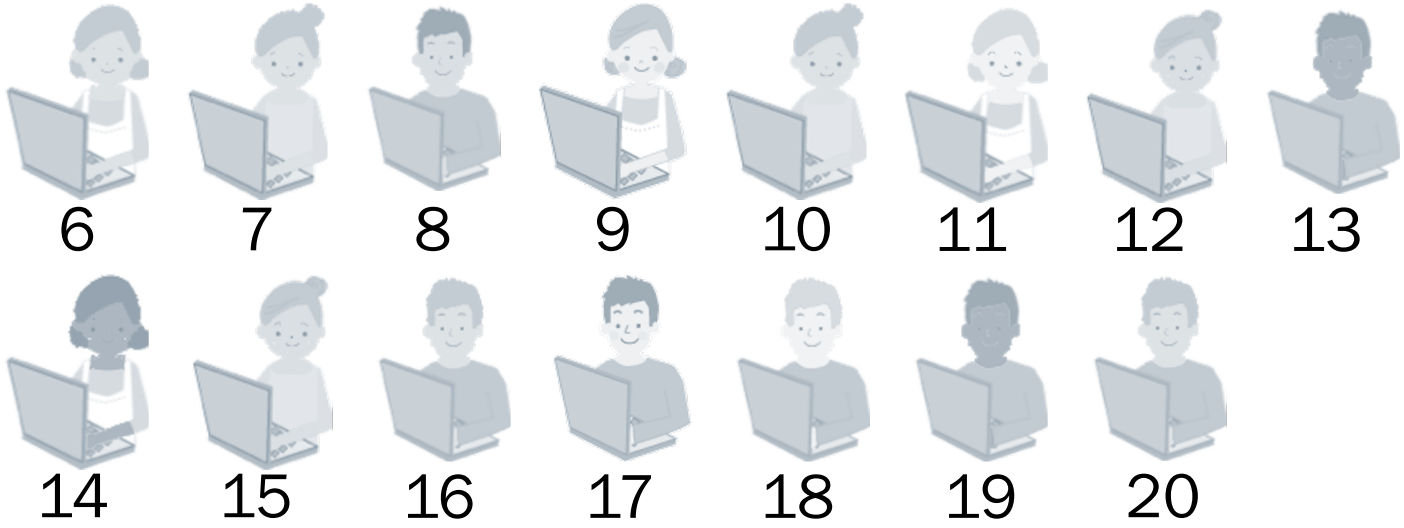
   $n!$ ways

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

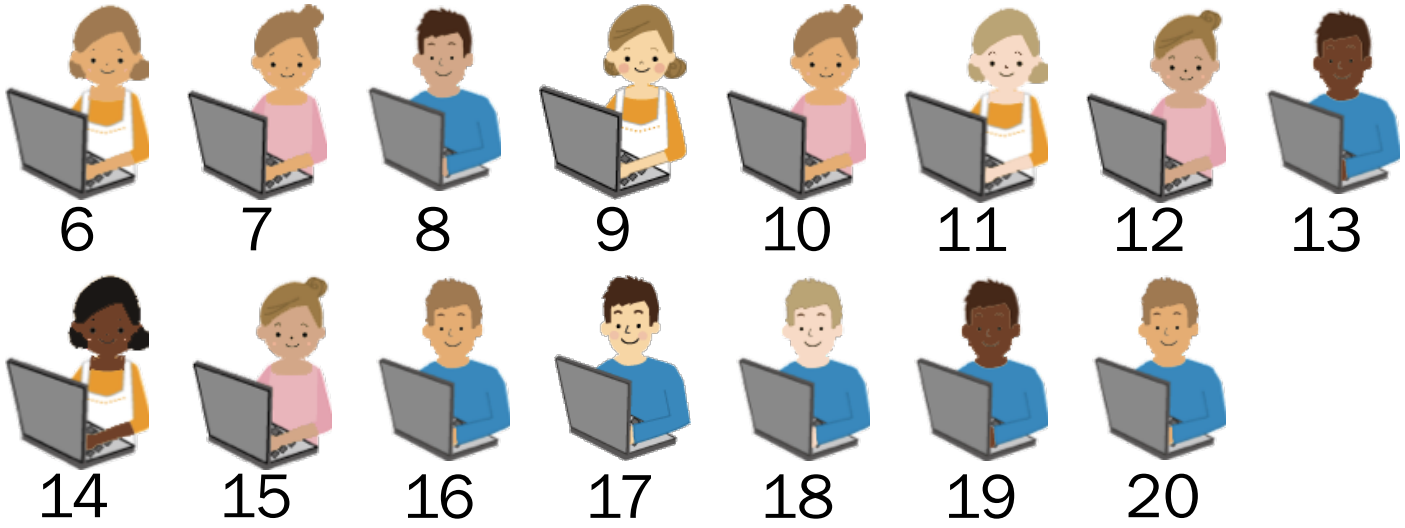
   1 way

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

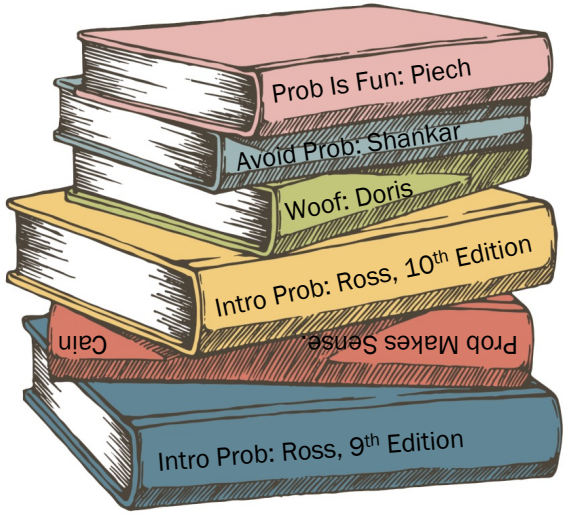
Fun Fact: $\binom{n}{k} = \binom{n}{n-k}$

$(x+y)^n$

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!

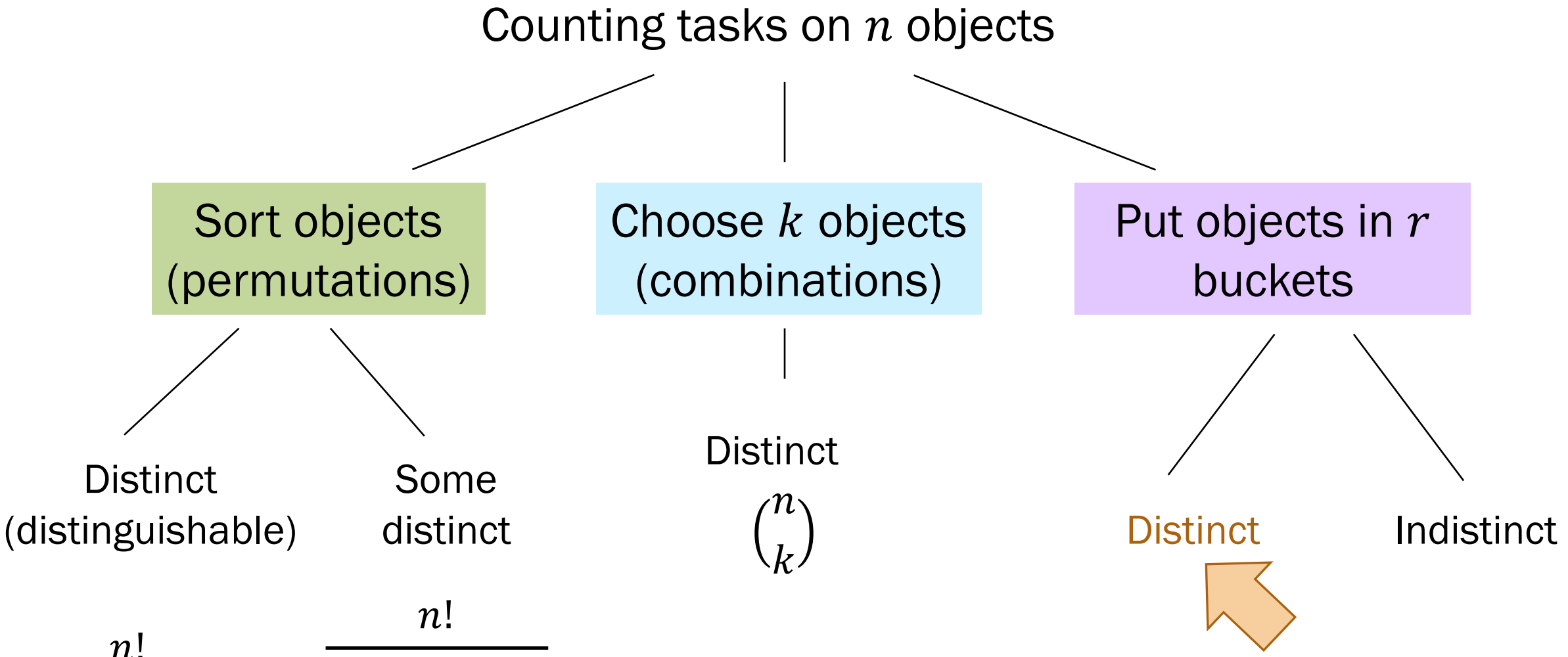
How many unique hands of 5 cards are there in a 52 card deck?

```
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)
```

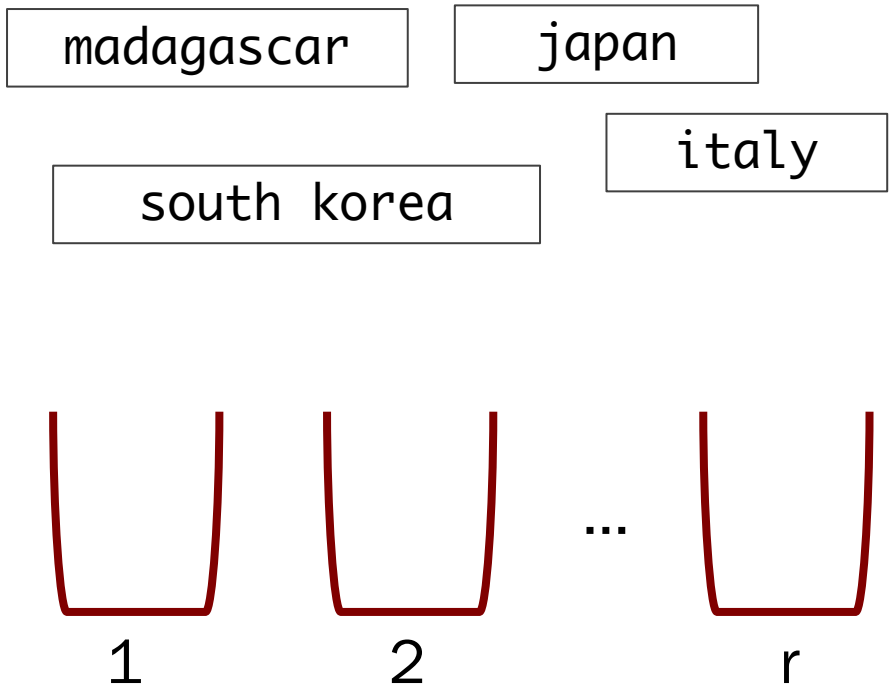
Buckets and The Divider Method

Summary of Combinatorics



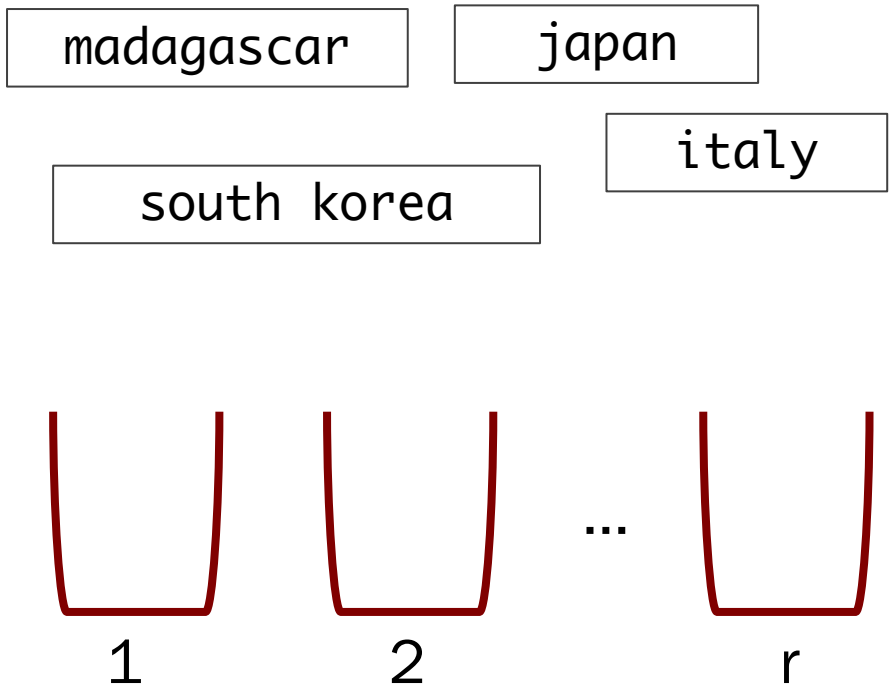
~~Balls and urns~~ Hash tables and distinct strings

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



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

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



Steps:

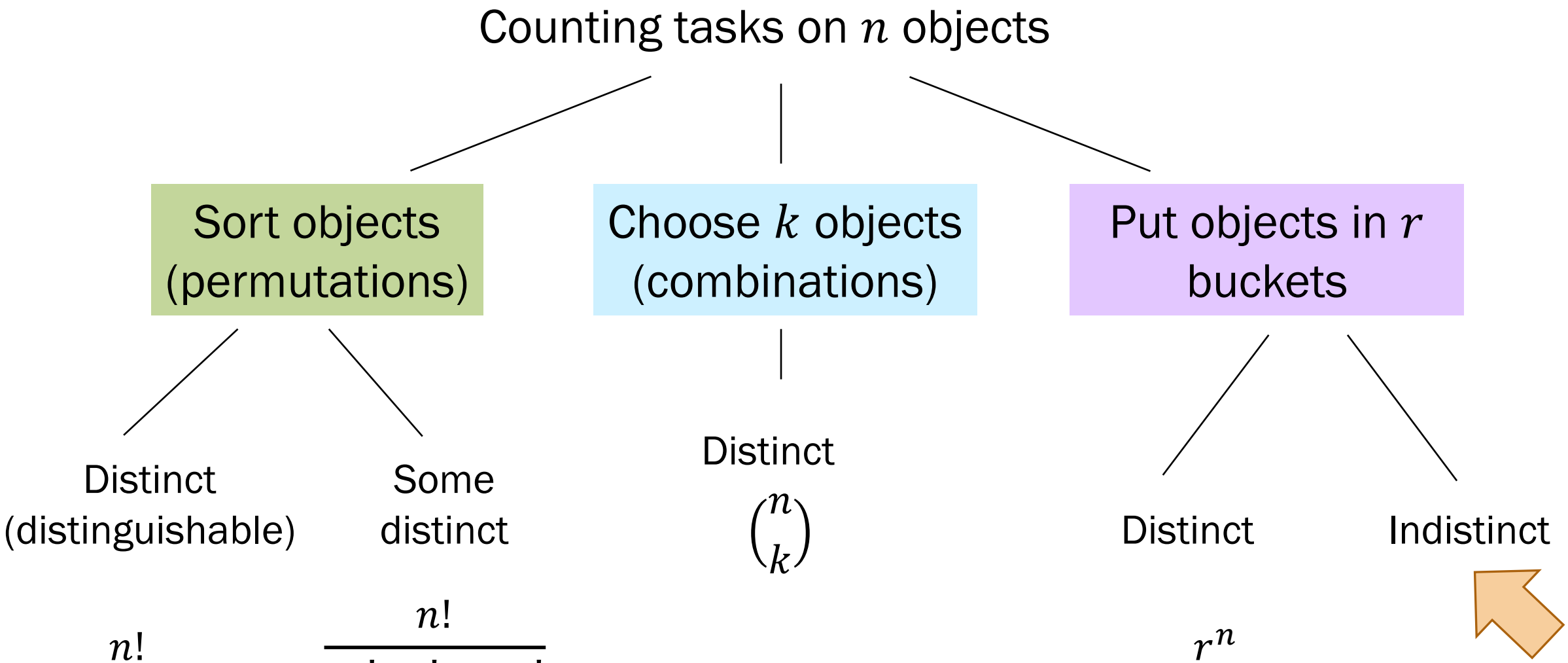
1. Bucket 1st string
2. Bucket 2nd string

...

n. Bucket n^{th} string

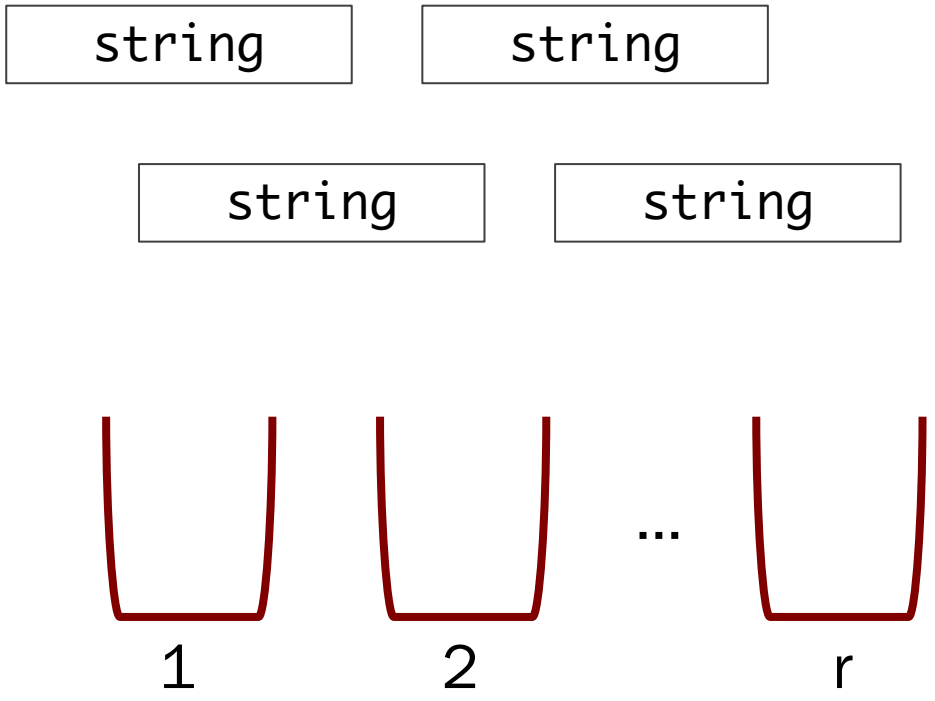
r^n outcomes

Summary of Combinatorics



Servers and indistinct requests

How many ways are there to distribute n indistinct strings to r buckets?



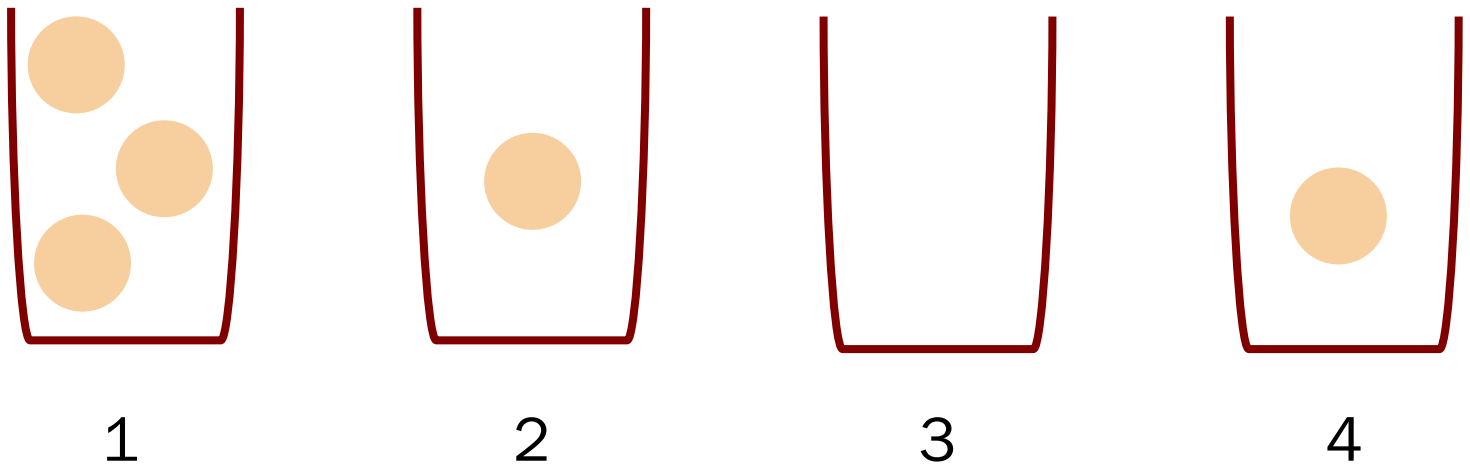
Goal

Bucket 1 has x_1 `string`,
Bucket 2 has x_2 `string`,
...
Bucket r has x_r `string`

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

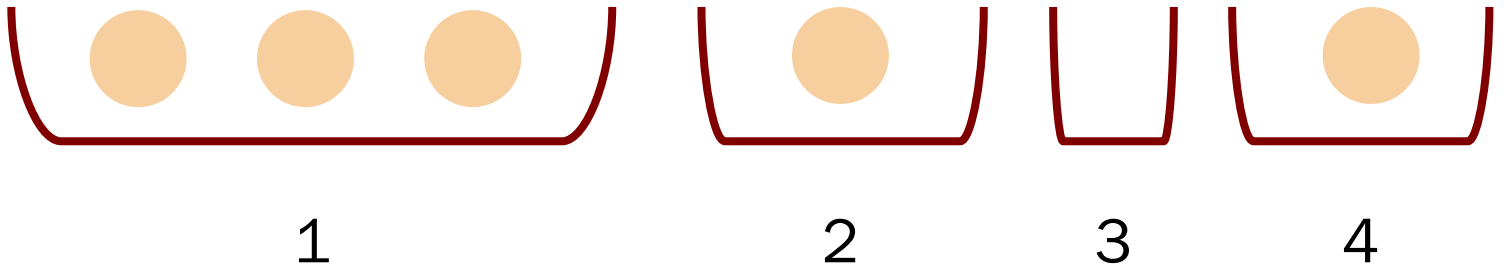
Servers and indistinct requests

How many ways are there to distribute n indistinct strings to r buckets?



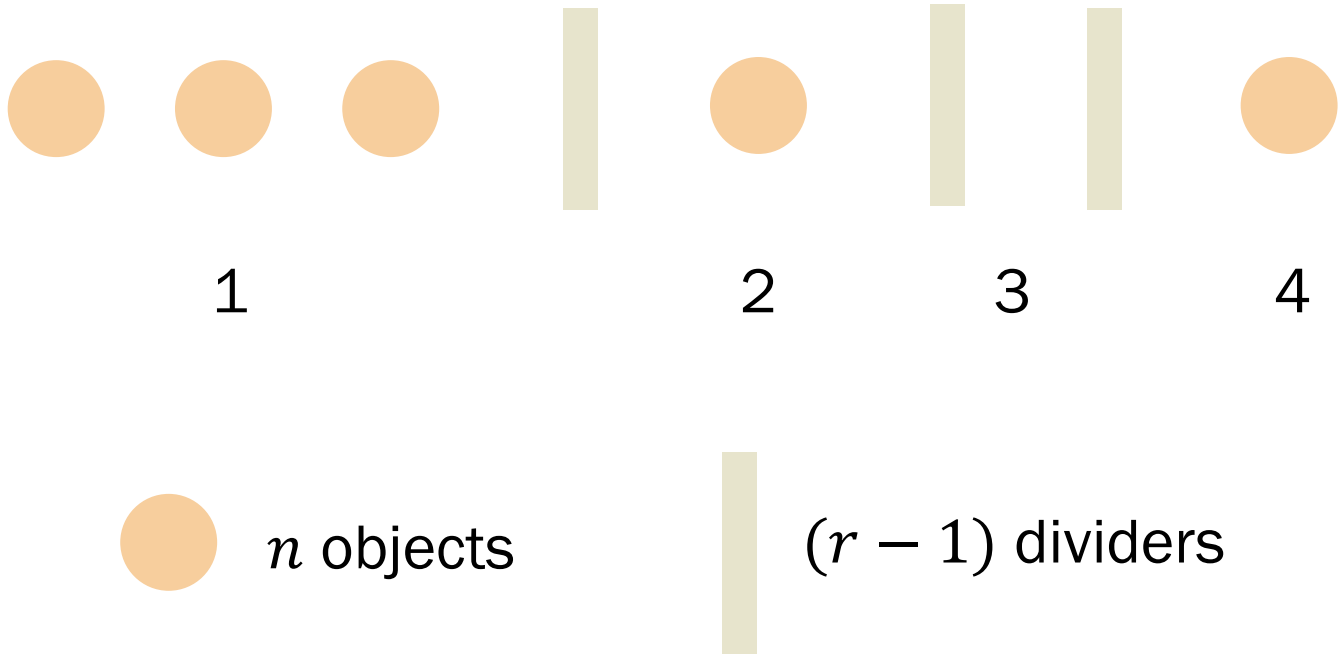
Servers and indistinct requests

How many ways are there to distribute n indistinct strings to r buckets?



Servers and indistinct requests

How many ways are there to distribute n indistinct strings to r buckets?



Servers and indistinct requests

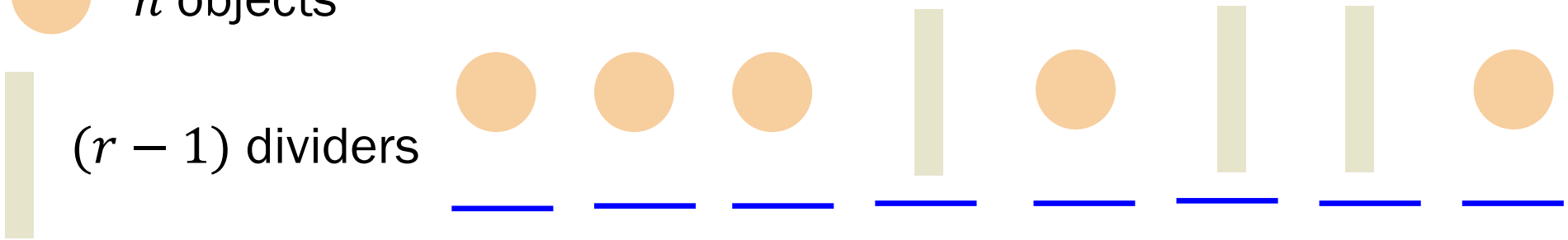
How many ways are there to distribute n indistinct strings to r buckets?

: Slots where either a divider or an object can go into

$(n + r - 1)$ slots

n objects

$(r - 1)$ dividers

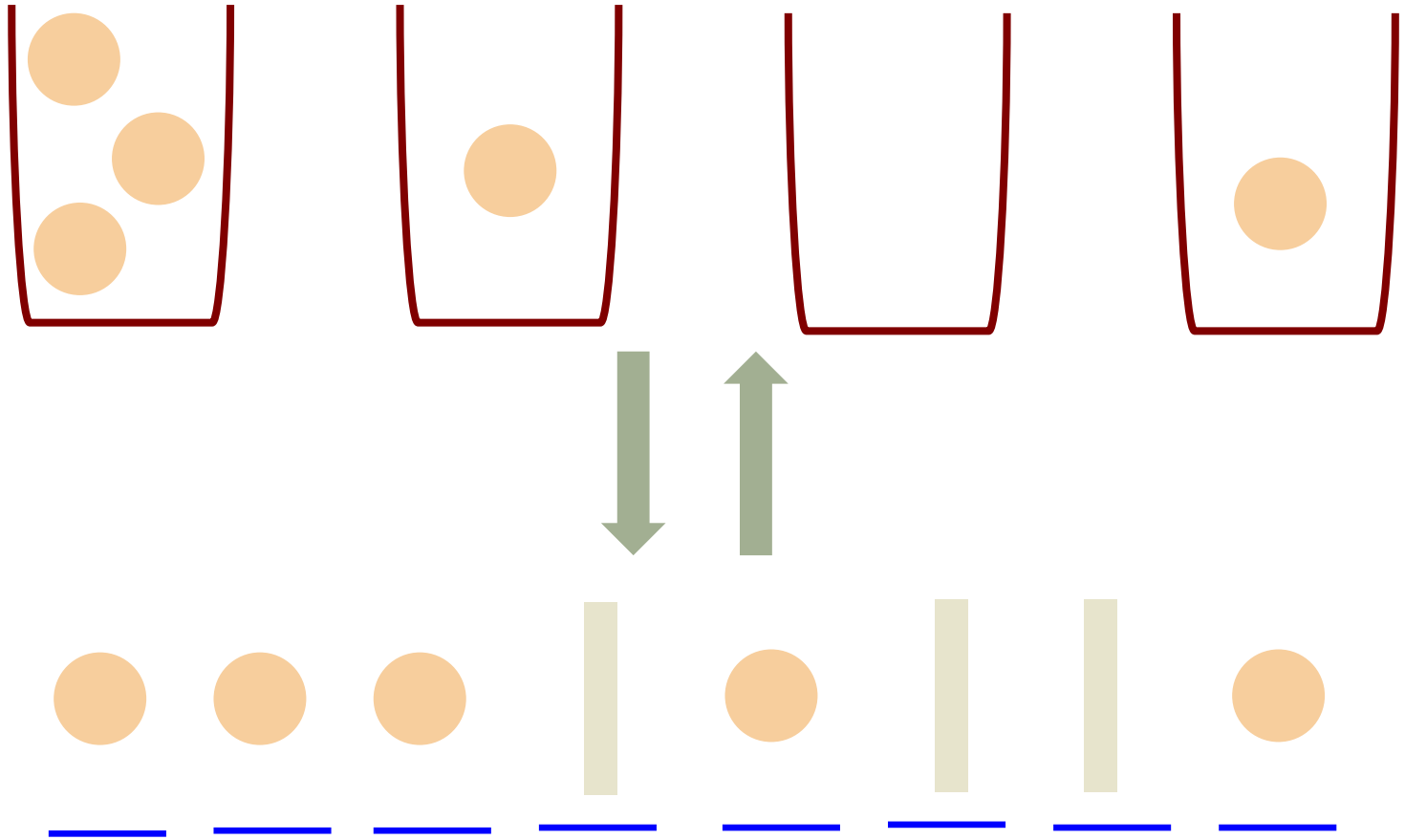


Servers and indistinct requests

How many ways are there to distribute n indistinct strings to r buckets?

n objects

r buckets



n objects

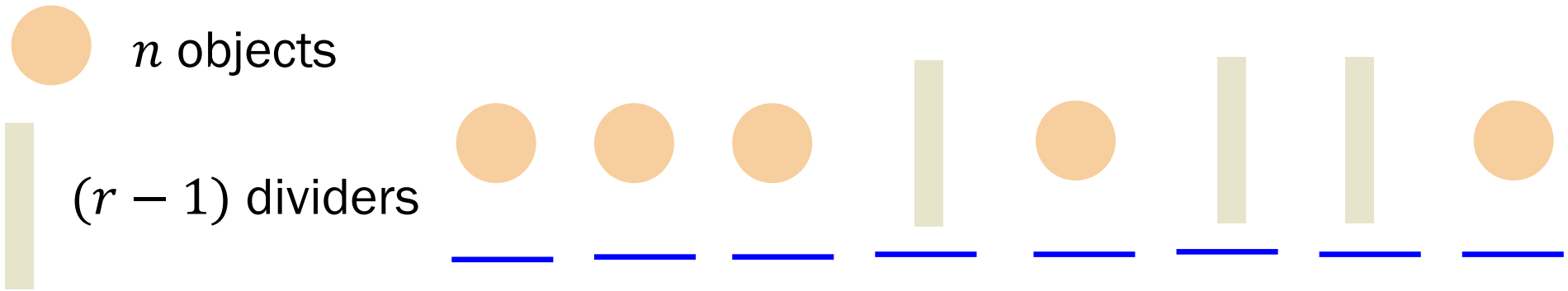
$(r-1)$ dividers

Servers and indistinct requests

How many ways are there to distribute n indistinct strings to r buckets?

____ : Slots where either a divider or an object can go into

$(n + r - 1)$ slots



Answer

1. (Combination) Choose $(r - 1)$ divider slots out of $(n + r - 1)$ distinct slots

2. (Permutation) Arrange $(r - 1)$ indistinct dividers and n indistinct objects

The divider method

The number of ways to distribute n indistinct objects into r buckets is equivalent to the number of ways of:

1. Choosing (r-1) slots among (n+r-1) distinct slots to place the divider
2. Arranging (r-1) indistinct dividers and n indistinct objects.

$$\text{Total} = \binom{n + r - 1}{r - 1} = \frac{(n + r - 1)!}{n!\,(r - 1)!} \quad \text{outcomes}$$

Integer solutions to equations

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

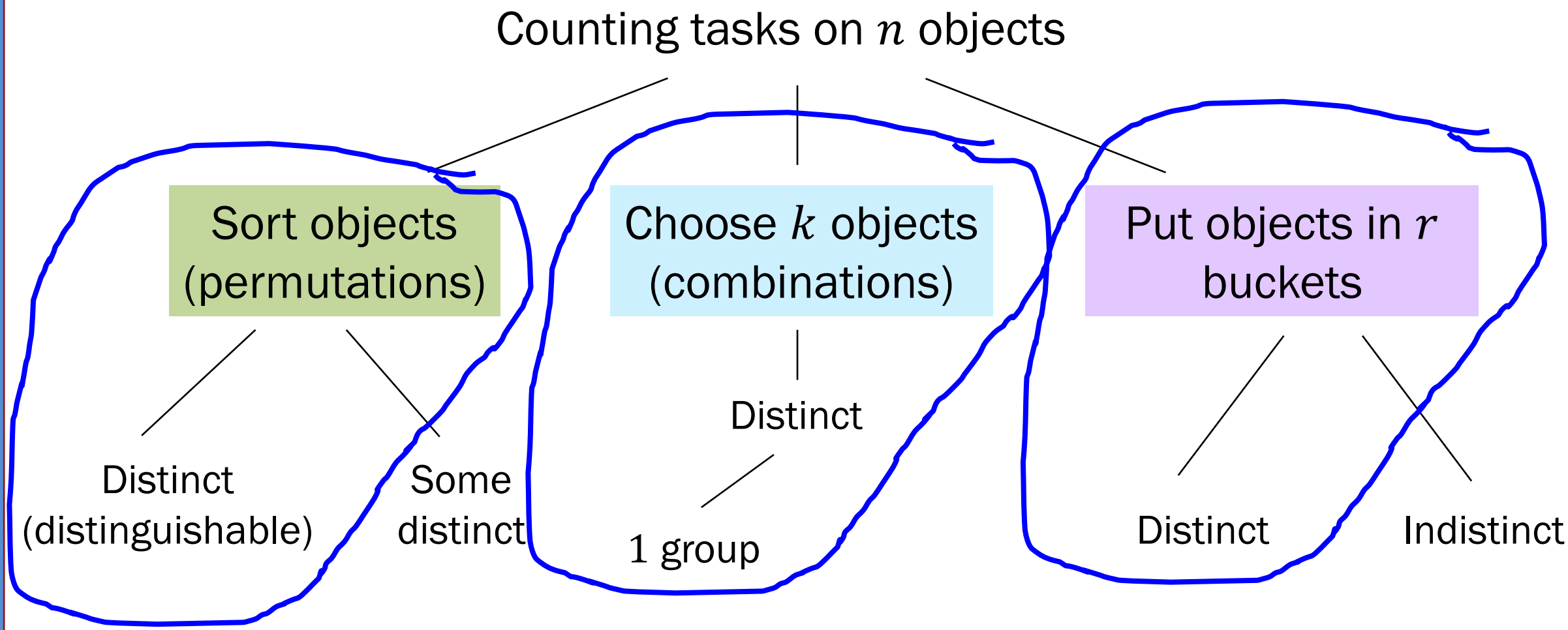
How many integer solutions are there to the following equation:

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

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



To the Code!

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

```python
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)
```

```python
import math

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

To the code!

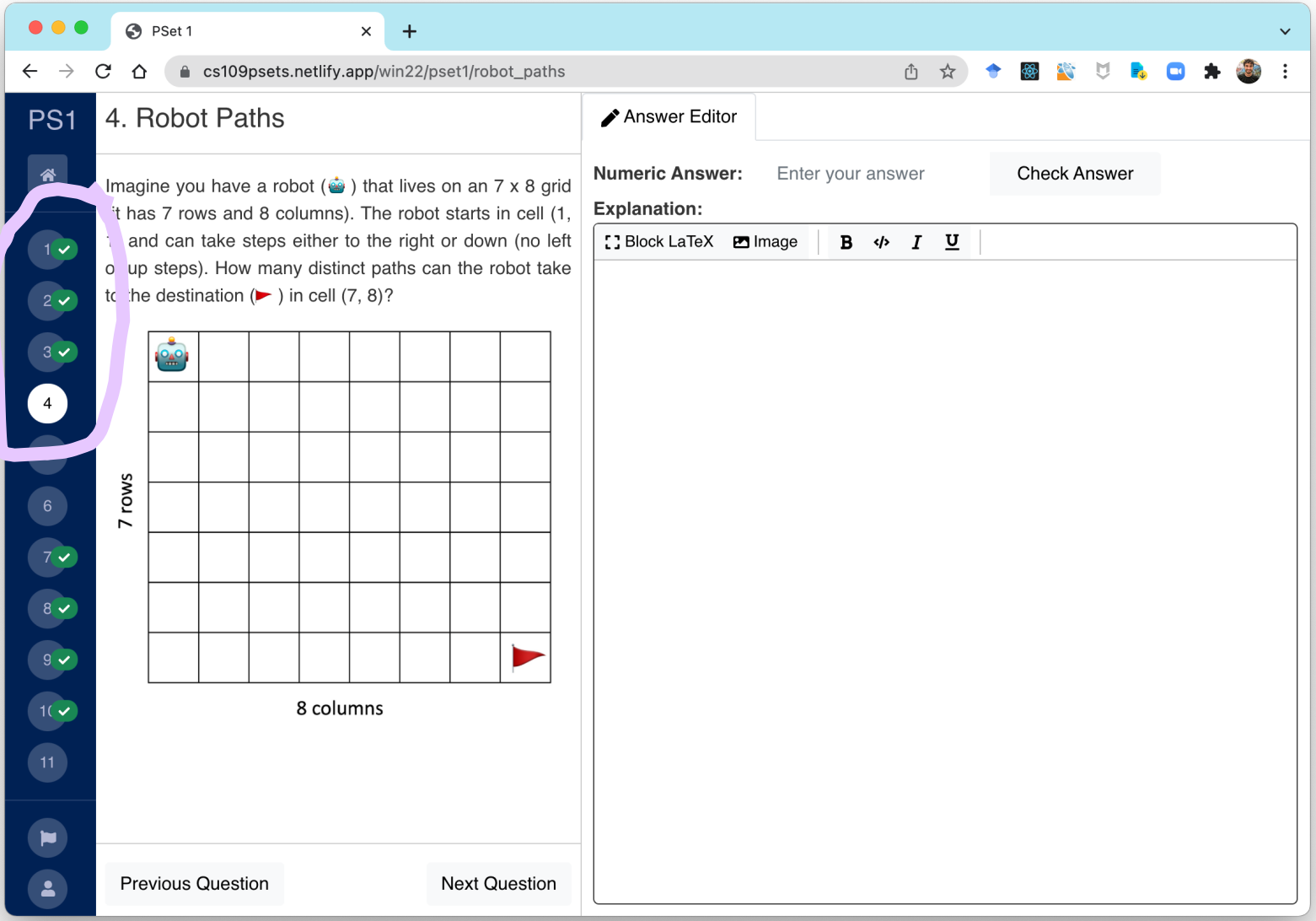
How many unique hands of 5 cards are there in a 52 card deck?

```
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)
```

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/

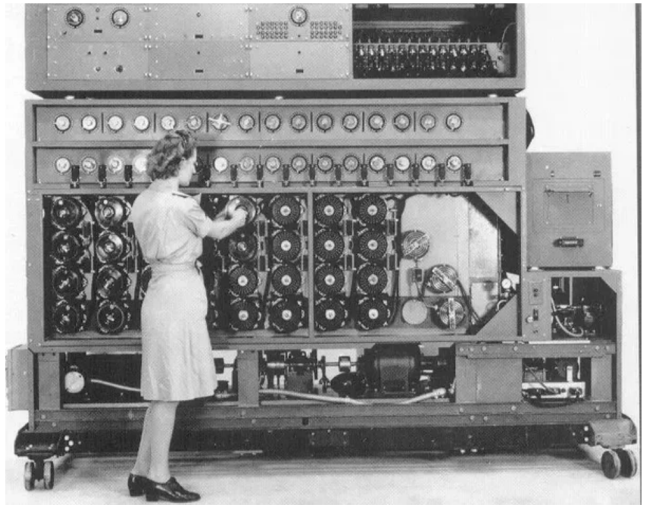
Update

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