combra.cs.rutgers.edu

Bring a notepad (it will become handy)

206
Discrete Structures II

Konstantinos P. Michmizos

Computational Brain Lab

Computer Science | Rutgers University | NJ, USA

Lecture 2 | 205 Recap | Wednesday September 13th 2023

What we will cover today and Tuesday

- Recap
  - How to do well in the course?
- Combinatorics Intro (Recap 205)
  - Sets
  - Venn Diagram
  - Functions
  - Proofs
  - Induction

Recap

How to do well in the course?

- Attend lectures and ask questions
  - There are no stupid questions
- Attend recitations
  - Will introduce new material
  - Will involve problem solving similar to quizzes/midterm
- Form study groups
  - Find a study buddy
- Come to office hours **prepared**
- Stay up to date with the material
  - Studying the day before a quiz/midterm is a bad idea.

How to do well in the course?

- Stay up to date with the material
  - Studying the day before a quiz/midterm is a bad idea
  - After each lecture the slides will be posted on canvas
  - Before the end of the day spend ~20 mins going over slides and make notes of things you did not understand
  - Bring questions to next class and office hours
  - When you attend a lecture, have a notepad in front of you (not a PC!!!) and solve (by hand) the problems (till the very end)

Recap

How to do well in the course?

- Lecture format
  - Each lecture will consist of introducing a concept and doing examples related to that concept
  - Extra Problems, quizzes, recitations, study groups will give you more practice on examples related to these concepts
  - The more you practice, the better you will get
    - It does not really matter what we will cover but what you will discover

Combinatorics

- The study of arrangements of objects

- Studied as long ago as the 17^{th} century, when combinatorial questions arose in the study of gambling games

- Used to solve many different types of problems
  - Examples:

  Enumeration, the **counting of objects** ***with certain properties***

  1. Counting determines the complexity of algorithms
  2. Counting determines whether there are enough resources to solve a problem
  3. …

Used to solve many different types of problems

Enumeration, the **counting of objects** ***with certain properties***

Example:

1. Counting determines the complexity of algorithms
2. Counting determines whether there are enough resources to solve a problem.

Combinatorics

- Study of discrete structures
  - Counting structures of a given kind/size

```
function TARJAN(Node* node)
    node.visited ← true
    node.index ← indexCounter
    s.push(node)
    for all successor in node.successors do
        if !node.visited then TARJAN(successor)
        end if
        node.lowlink ← MIN(node.lowlink, successor.lowlink)
    end for
    if node.lowlink == node.index then
        repeat
            successor ← stack.pop()
        until successor == node
    end if
end function
```

What questions can you ask?

Combinatorics

Used to solve many different types of problems

Enumeration, the **counting of objects** ***with certain properties***

Example:

1. Counting determines the complexity of algorithms
2. Counting determines whether there are enough resources to solve a problem.

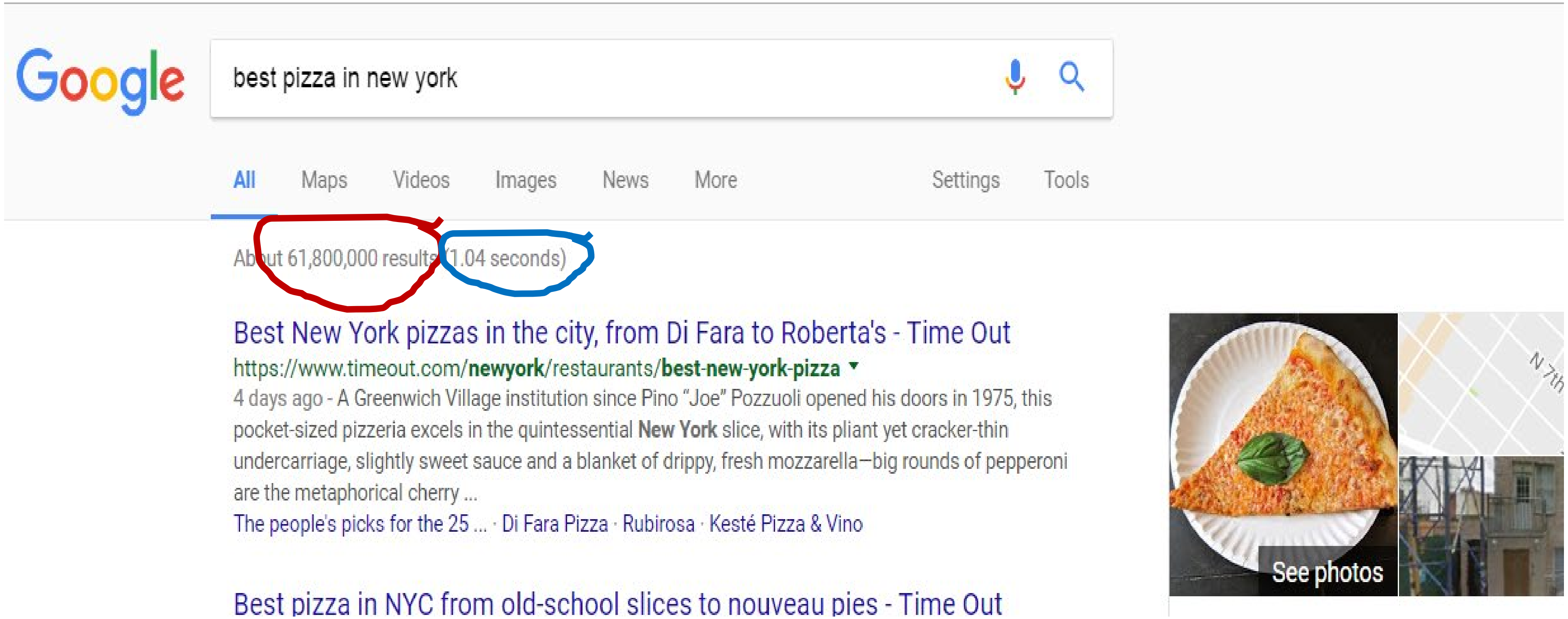
- Study of discrete structures
  - Counting structures of a given kind/size

```
function TARJAN(Node* node)
    node.visited ← true
    node.index ← indexCounter
    s.push(node)
    for all successor in node.successors do
        if !node.visited then TARJAN(successor)
        end if
        node.lowlink ← MIN(node.lowlink, successor.lowlink)
    end for
    if node.lowlink == node.index then
        repeat
            successor ← stack.pop()
        until successor == node
    end if
end function
```

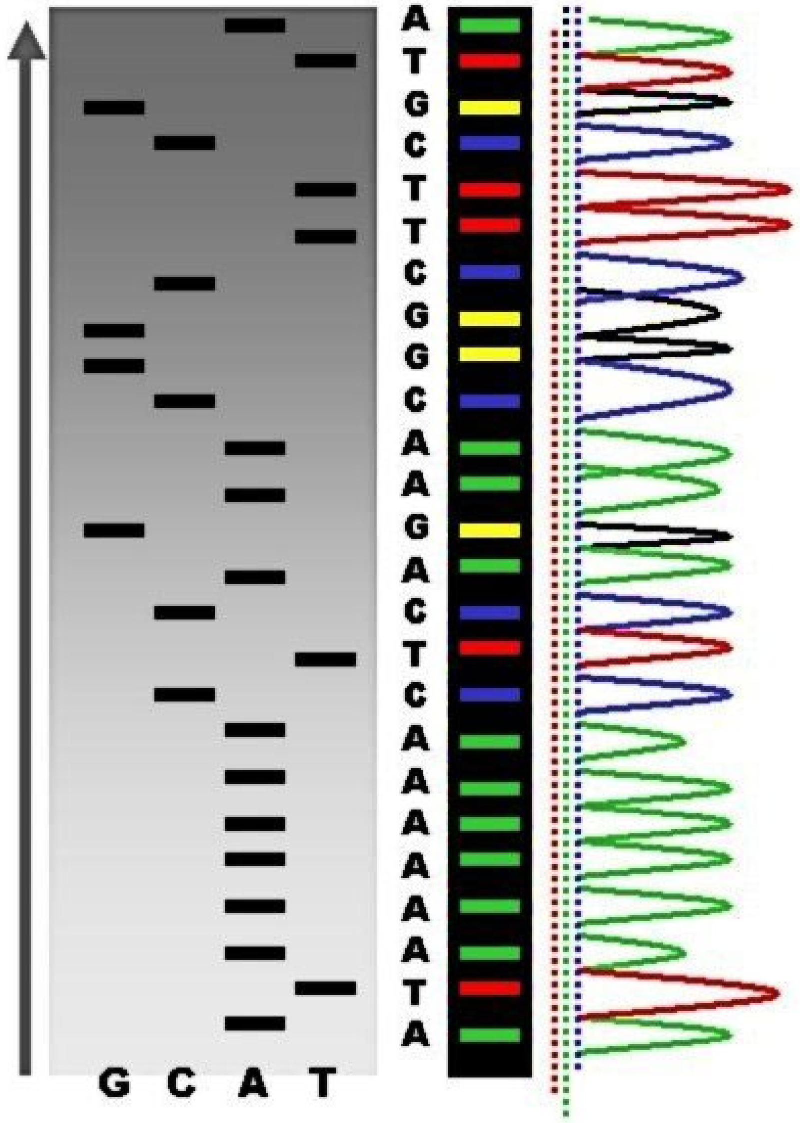
Complexity:
What is the runtime?

Resources:
What is the memory usage?

Combinatorics – Enumerating Example



Combinatorics



Recently, it has played a key role in

mathematical biology,

e.g., in sequencing DNA.

Combinatorics

- We will study the **basic rules of counting**
  - They can solve a tremendous variety of problems, such as:
    - Enumerate the **different telephone numbers** possible in the United States,
    - Enumerate the **allowable passwords** on a computer system,
    - Enumerate the **different orders** in which the runners in a race can finish
  - They can help us answer questions that seem hard: *What is the chance that among the 150 students in this class, we find 2 with the same birthday?*

- An important **combinatorial tool** is the *pigeonhole principle*: When objects are placed in boxes and there are more objects than boxes, then there is a box containing at least 2 objects.
  - E.g., we can use this principle to show that among a set of 15 or more students, at least 3 were born on the same day of the week

Combinatorics

Your Password:

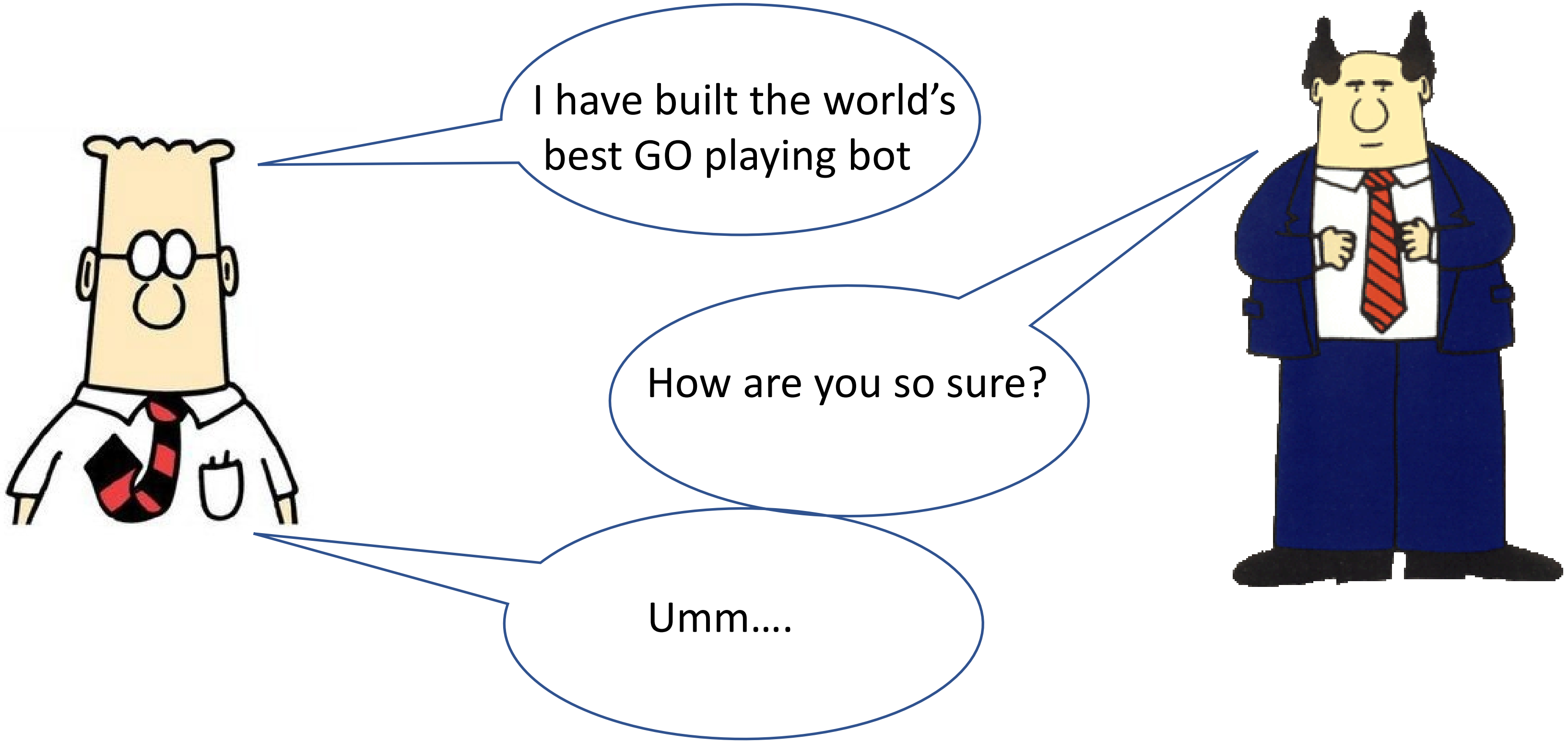
- Must be different from your User ID
- Must contain 8 to 20 characters, including one letter and number
- May include one of the following characters: %, &, _, ?, #, =, -
- Your new password cannot have any spaces and will not be case sensitive.

*REQUIRED FIELD

How many different passwords can we create?

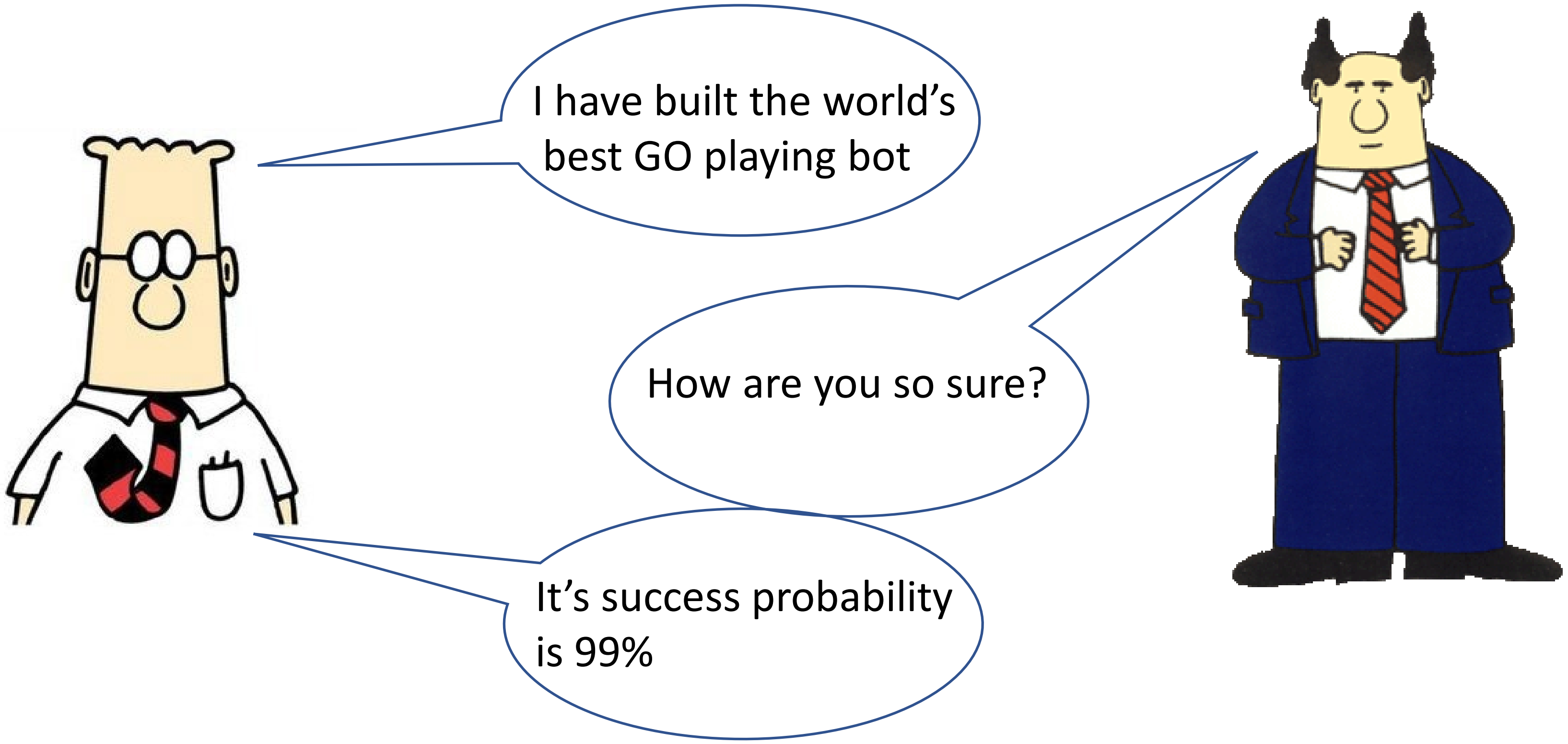
Combinatorics ---> Probability Theory

- Analysis of uncertain or random phenomena



Probability Theory

- Analysis of uncertain or random phenomena



Probability Theory

- Analysis of uncertain or random phenomena

Probability Theory

- Analysis of uncertain or random phenomena

Course Outline

- Part I
  - Recap of basics – sets, function, proofs, induction
  - Basic counting techniques
  - Pigeonhole principle
  - Generating functions
- Part II
  - Sample spaces and events
  - Basics of probability
  - Independence, conditional probability
  - Random variables, expectation, variance
  - Moment generating functions
- Part III
  - Graph Theory
  - Machine learning and statistical inference

Lehman, 69-72

Sets

- What is a *Set?*
  - A collection of objects which are called *elements*
  - Elements are objects that share the same property
- Examples
  - Followers on Twitter
  - The set of webpages for a given Google query
  - Collection of YouTube videos

Sets

- The order of elements is not significant, so $\{x, y\}$ and $\{y, x\}$ are the same set written two different ways.

- And what about $y = x$?
  - $\{x, x\} = \{x\}$

- The expression $e \in S$ asserts that e **is an element of** set S
  - E.g.,$32 \in S$ or $blue \notin S$

Sets – Common Sets

- What is a *Set?*
  - A collection of objects which are called *elements*.
- Some common sets in Math

curly braces

$\emptyset$	Empty set	{}
$\mathbb{N}$	Nonnegative integers	{0,1,2,3,…}
$\mathbb{Z}$	Integers	{…,-2,-1,0,1,2,…}
$\mathbb{Q}$	Rational numbers	{1/2, 16, -5/2}
$\mathbb{R}$	Real numbers	$\{\pi, e, -9, \sqrt{2}\}$
$\mathbb{C}$	Complex numbers	$\{i, 19/2, \sqrt{2}\text{-}2i\}$

A superscript "+" restricts a set to its positive elements; for example, $\mathbb{R}^+$ denotes the set of positive real numbers. Similarly, $\mathbb{Z}^-$ denotes the set of negative integers

For example
$X ::= \{1,2,3\}$
$Y ::= \{2,3,4\}$

Sets - Set Operations

- Union: $X \cup Y$
  - All elements present in X or Y or both.

$X \cup Y = \{1,2,3,4\}$

- Intersection: $X \cap Y$
  - All elements present in *both* X and Y.

$X \cap Y = \{2,3\}$

- Difference: $X \setminus Y$
  - All elements present in X but not in Y.
  - Not symmetric!

$X \setminus Y = \{1\}$
$Y \setminus X = \{4\}$

- Product: $X \times Y$
  - Collection of all tuples (a, b) where $a \in X$ and $b \in Y$.

$X \times Y = \{(1,2),(1,3),(1,4),(2,2),(2,3),(2,4),(3,2),(3,3),(3,4)\}$

- Size: $|X|$
  - Number of elements in X.

$|X| = 3$
$|Y| = 3$

Sets - Set Comparisons

- Subset: $X \subset Y$
  - Every element present in X is also present in Y.
  - X is not the same as Y.

$X = \{1\}, Y = \{1, 2, 3\}$

$X \subset Y$

- Superset: $X \supset Y$
  - Every element present in Y is also present in X.
  - X is not the same as Y.

- Note: There is a direct analogy between [$\subset$ and $<$] and [$\subseteq$ and $\leq$]

Power Set

$X = \{1,2,3\}$
$Power(X) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$

- Let X be a set.
- $Power(X)$ = set of all subsets of X

- E.g., $Power(\{1,2\}) = \{1\}, \{2\}, and\ \{1,2\}$

- Is this correct?
  - NO!
  - $Power(\{1,2\}) = \{1\}, \{2\}, \{1,2\}, \boldsymbol{and}\ \{\}$

- Generally, if A has n elements, then there are 2^n sets in $Power(A)$

Set Builder Notation

- Often sets cannot be fully described by listing the elements explicitly or by taking unions, intersections, etc., of easily-described sets
- **Set builder notation** often comes to the rescue
- The idea is to define a set using a predicate; in particular, the set consists of all values that make the predicate true

Examples:

- $X = \{n \in \mathbb{N}: n\ is\ prime\}$
- $Y = \{x \in \mathbb{R}: x^3 - 3x + 1 > 0\}$
- $Z = \{z \in YouTube\_videos: z\ is\ less\ than\ 3\ minutes\ long\}$

Exercise 1: Put everything together

$A = \{0, 1, 2\}$ $B = \{1, 4, 9\}$

- Let $A = \{n \in N: n^2 < 7\}$ and $B = \{1,4,9\}$

Find

- $A \cup B$ $A \cup B = \{0, 1, 2, 4, 9\}$
- $A \cap B$ $A \cap B = \{1\}$
- $A \times B$ $A \times B = \{(0,1), (0,4) \ldots$
- $A \setminus B$ $A \setminus B = \{0, 2\}$

Venn Diagram

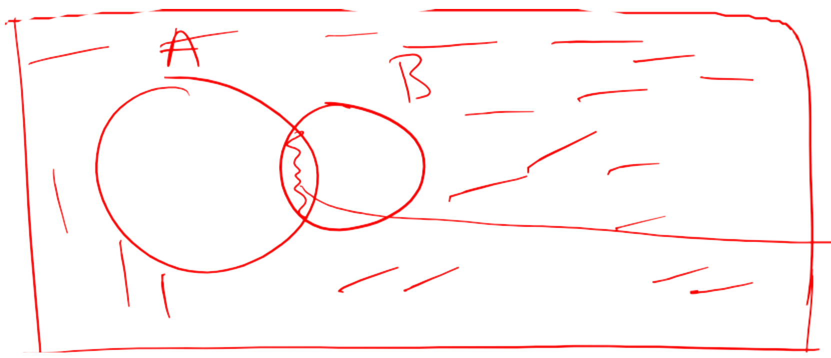
- Represent sets as circles and elements as points within it.
- Elegant way to capture relationships among sets.

Work on it

Exercise 2: Venn Diagram

- There are 131 students in CS 206.
- 100 like chocolate ice cream. 50 like vanilla ice cream.
- 20 like both chocolate and vanilla ice cream.
- Draw a Venn diagram to represent this.
- How many students do not like either flavor of ice cream.

A = chocolate
B = vanilla



$|A| = 100$
$|B| = 50$
$|A \cap B| = 20$