CIT 5920

Mathematical Foundations of Computer Science
Or how to communicate formally

FALL 2024

Jérémie Lumbroso, Ph.D.

Aug 27th, 2024

University of Pennsylvania

Computer & Information Science

Welcome/Agenda

- Welcome today!
- Get situated with **Sli.do**, your participation system
- Staff introductions
- Course syllabus: What is the point?
- Administrivia

Sli.do ACCESS

Audience participation with **Sli.do**

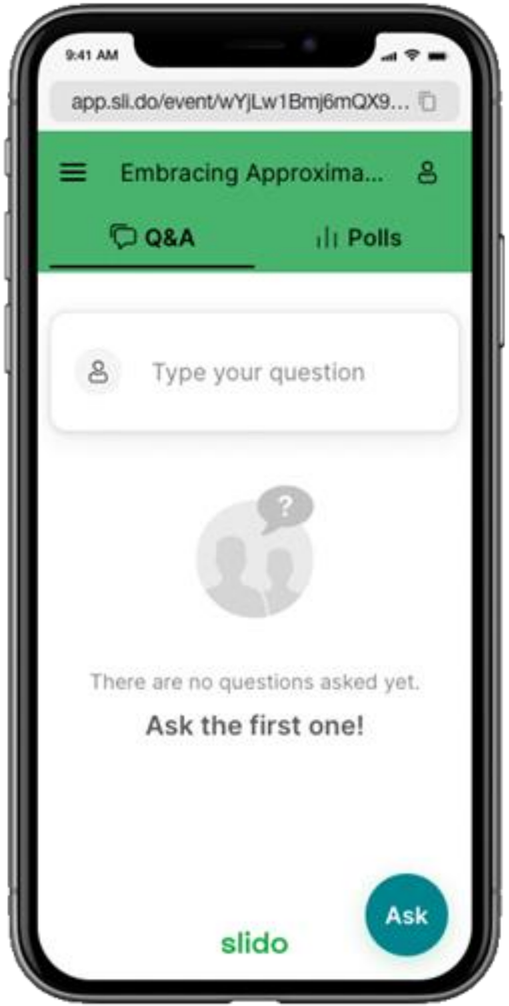
You can:

- **ask questions** and **upvote others' questions**
- **answer quizzes**
- choose to be anonymous (or not!)

Go to https://sli.do and enter the event code is #3017135

This should be a wonderful experience for you, please speak up!

slido.com with
#3017135



9:41 AM

app.sli.do/event/wYjLw1Bmj6mQX9...

Embracing Approxima...

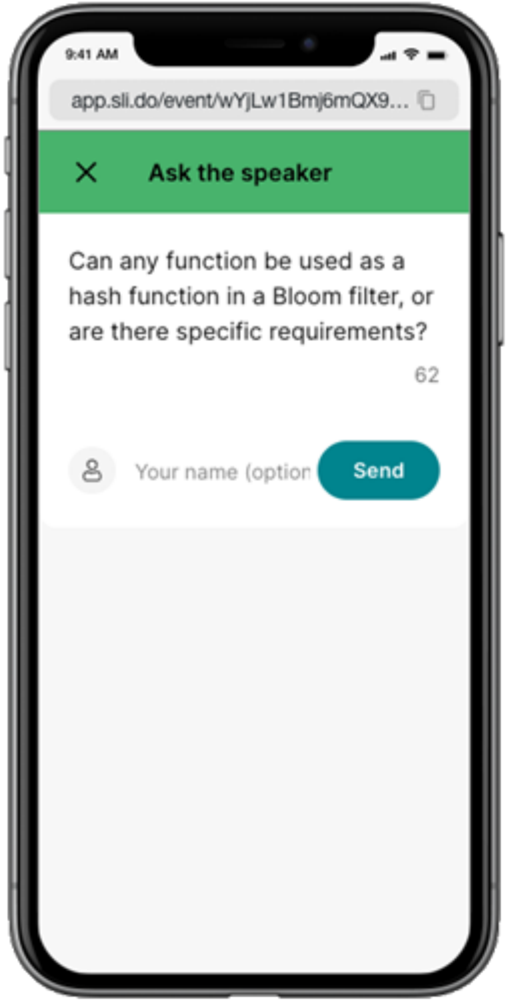
Q&A

Polls

Type your question

There are no questions asked yet.

Ask the first one!



9:41 AM

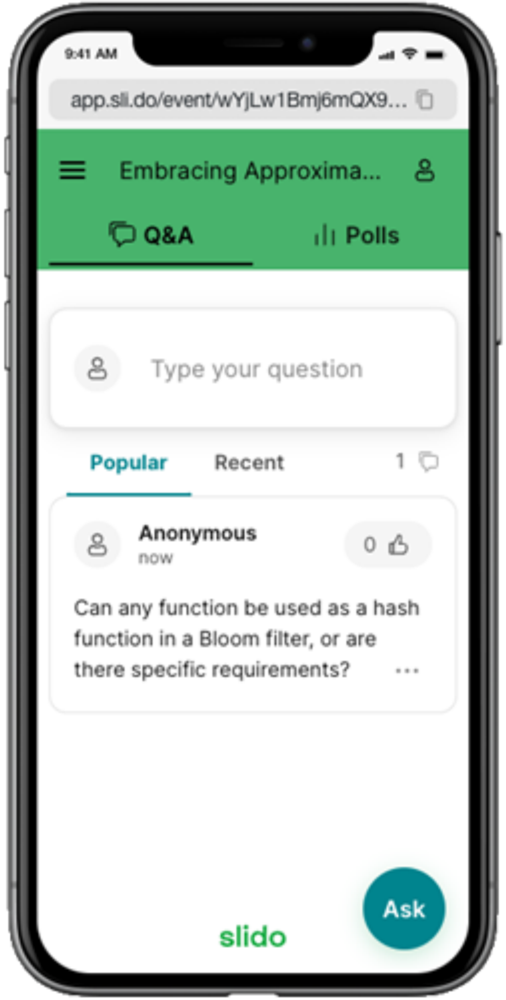
app.sli.do/event/wYjLw1Bmj6mQX9...

Ask the speaker

Can any function be used as a hash function in a Bloom filter, or are there specific requirements?

62

Your name (option



9:41 AM

app.sli.do/event/wYjLw1Bmj6mQX9...

Embracing Approxima...

Q&A

Polls

Type your question

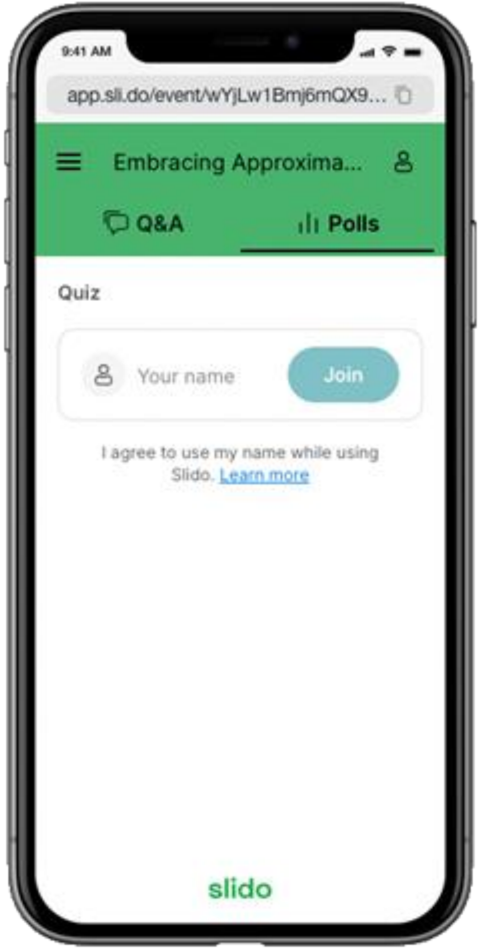
Popular Recent 1

Anonymous
now

0

Can any function be used as a hash function in a Bloom filter, or are there specific requirements?

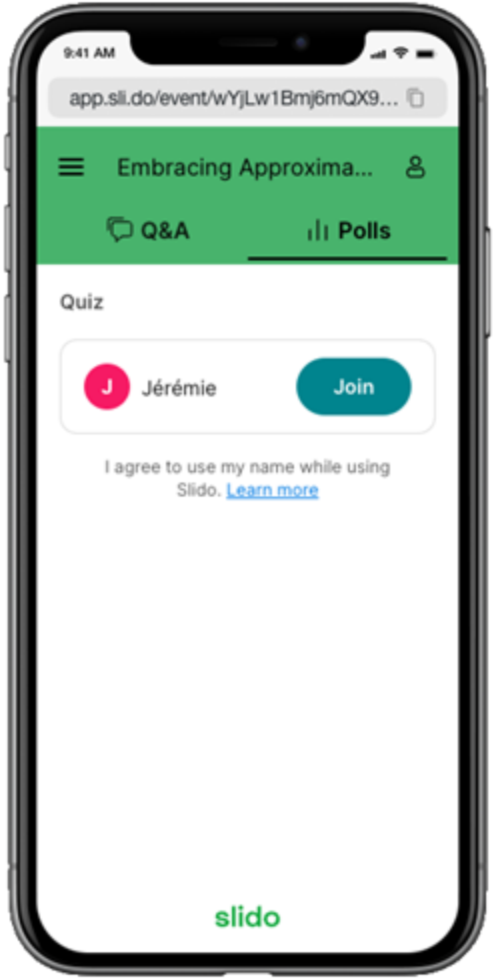
slido.com
with
#3017135



Quiz

Your name

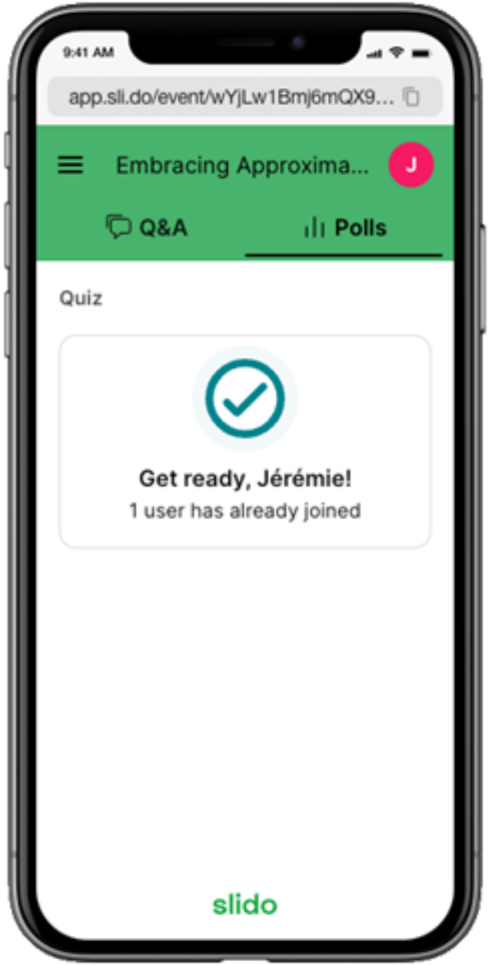
I agree to use my name while using Slido. Learn more



Quiz

Jérémie

I agree to use my name while using Slido. Learn more



Quiz

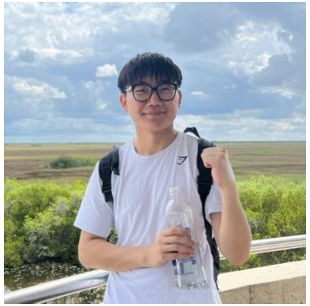
Get ready, Jérémie!
1 user has already joined

slido.com
with
#3017135

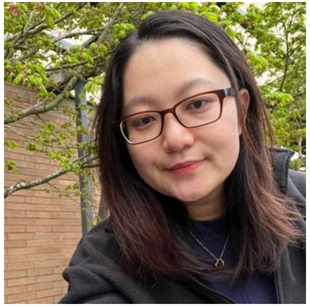
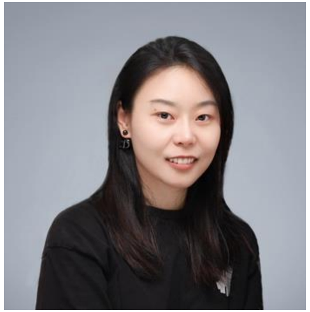
STAFF INTRODUCTIONS

TA staff

RECITATION TAs

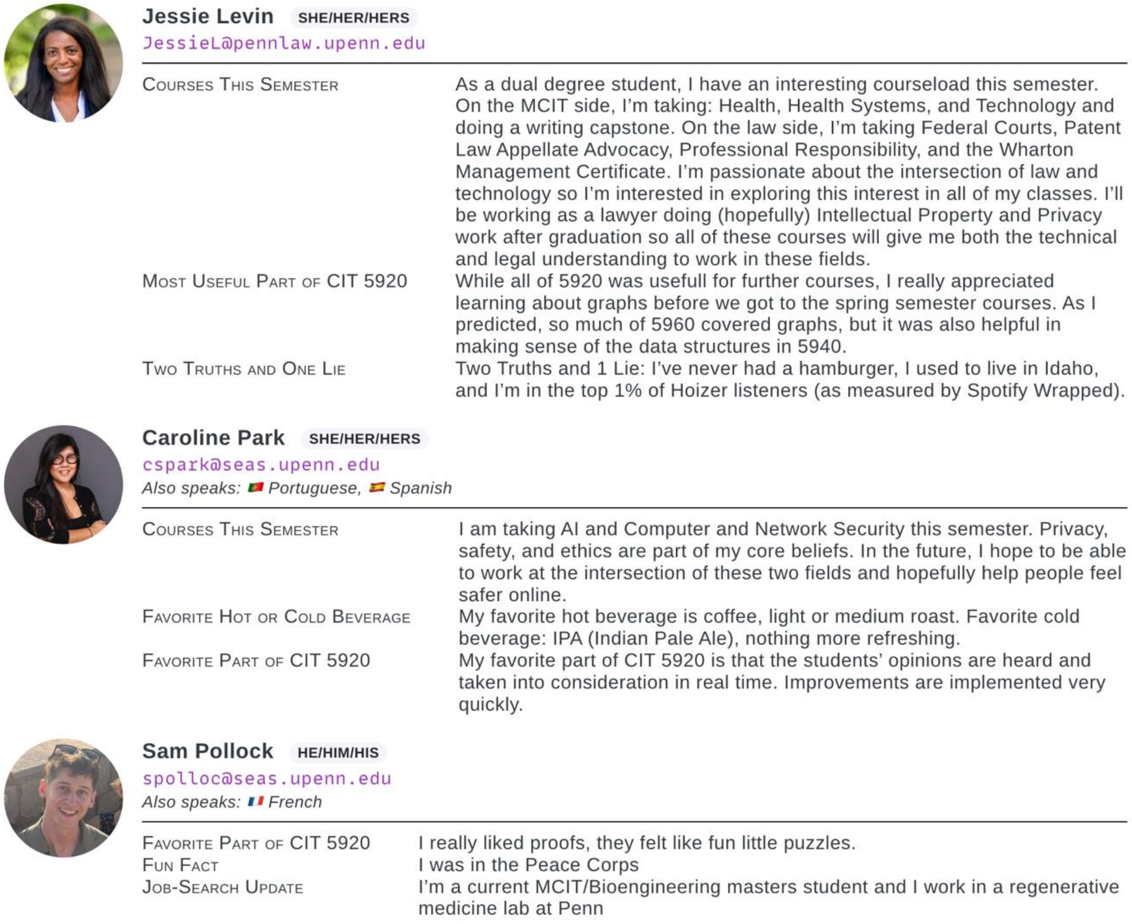


TAs

TA staff

- The website will contain much more information
  - Some ice breakers
  - Alternate languages spoken
- Will be updated with office hours by next week



Jessie Levin SHE/HER/HERS
JessieL@pennlaw.upenn.edu

Courses This Semester
As a dual degree student, I have an interesting courseload this semester. On the MCIT side, I'm taking: Health, Health Systems, and Technology and doing a writing capstone. On the law side, I'm taking Federal Courts, Patent Law Appellate Advocacy, Professional Responsibility, and the Wharton Management Certificate. I'm passionate about the intersection of law and technology so I'm interested in exploring this interest in all of my classes. I'll be working as a lawyer doing (hopefully) Intellectual Property and Privacy work after graduation so all of these courses will give me both the technical and legal understanding to work in these fields.

Most Useful Part of CIT 5920
While all of 5920 was usefull for further courses, I really appreciated learning about graphs before we got to the spring semester courses. As I predicted, so much of 5960 covered graphs, but it was also helpful in making sense of the data structures in 5940.

Two Truths and One Lie
Two Truths and 1 Lie: I've never had a hamburger, I used to live in Idaho, and I'm in the top 1% of Hoizer listeners (as measured by Spotify Wrapped).

Caroline Park SHE/HER/HERS
cspark@seas.upenn.edu
Also speaks: 🇵🇹 Portuguese, 🇪🇸 Spanish

Courses This Semester
I am taking AI and Computer and Network Security this semester. Privacy, safety, and ethics are part of my core beliefs. In the future, I hope to be able to work at the intersection of these two fields and hopefully help people feel safer online.

Favorite Hot or Cold Beverage
My favorite hot beverage is coffee, light or medium roast. Favorite cold beverage: IPA (Indian Pale Ale), nothing more refreshing.

Favorite Part of CIT 5920
My favorite part of CIT 5920 is that the students' opinions are heard and taken into consideration in real time. Improvements are implemented very quickly.

Sam Pollock HE/HIM/HIS
spolloc@seas.upenn.edu
Also speaks: 🇫🇷 French

Favorite Part of CIT 5920
I really liked proofs, they felt like fun little puzzles.

Fun Fact
I was in the Peace Corps

Job-Search Update
I'm a current MCIT/Bioengineering masters student and I work in a regenerative medicine lab at Penn

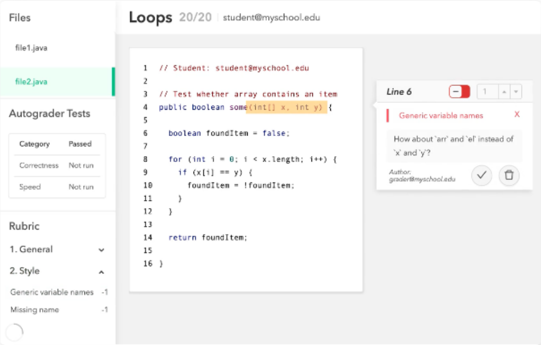
Expectations for TAs

- TAs have all been hand-recruited by me, the instructor
- Head TA: TBA
- Expectations
  - They know a lot, and what they do not know they know how to verify
  - You can trust TAs as **the most reliable resource** of the course
  - Their job is to care and be empathetic
  - Many of them have gone through exactly the course you are going through, and they both want to help and know how to
- **A lot of the best ideas come from TAs**
- Engage with them often!
- **You will see them on Friday in Recitation 1**

Jérémie Lumbroso

- Originally from France
- MS, Ph.D from **Sorbonne Université**
- Has taught in
  - Simon Fraser University (Vancouver, 2013-2014)
  - Princeton University (2014-2023)
- Has created startups in education
  - codePost
  - CUvids

Probabilistic Algorithms for Data Streaming and Random Generation



Loops 20/20 | student@myschool.edu

Category	Passed
Correctness	Not run
Speed	Not run

Generic variable names -1

Missing name -1

```
1  // Student: student@myschool.edu
2
3  // Test whether array contains an item
4  public boolean some(int[] x, int y) {
5
6    boolean foundItem = false;
7
8    for (int i = 0; i < x.length; i++) {
9      if (x[i] == y) {
10       foundItem = !foundItem;
11     }
12   }
13
14   return foundItem;
15
16 }
```

Line 6

Generic variable names

How about `arr` and `el` instead of `x` and `y`?

Author: grader@myschool.edu

cuvids

SORBONNE UNIVERSITÉ

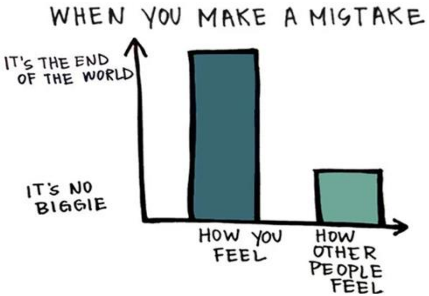
My teaching philosophy

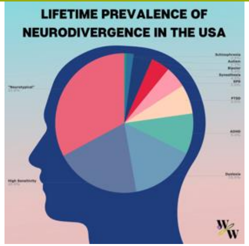
- Your success is my top priority
- We are here **for you**
- I focus on trying to get you to engage with the material
- I prioritize student questions
- *I try to make grading a secondary matter!*
- I try to make sure I am clear

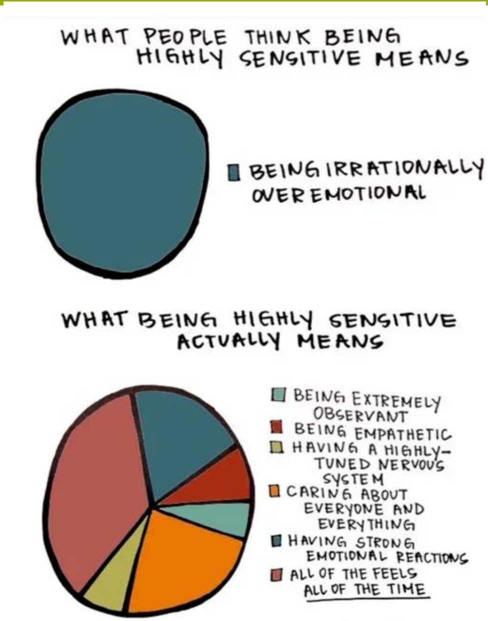
- YOU are the superstar, YOU are each other’s best assets!

- **What I admire the most**
  - Kindness
  - Helping others
  - Patience

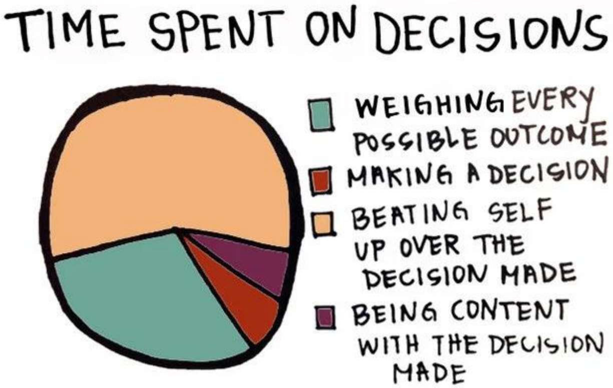
Neurodivergence





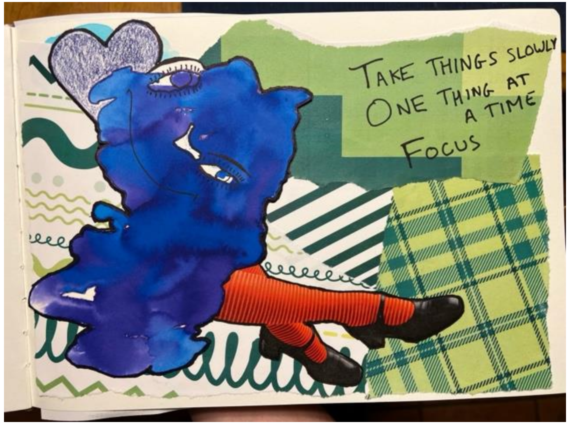

WHAT PEOPLE THINK BEING HIGHLY SENSITIVE MEANS
BEING IRRATIONALLY OVEREMOTIONAL
WHAT BEING HIGHLY SENSITIVE ACTUALLY MEANS
BEING EXTREMELY OBSERVANT
BEING EMPATHETIC
HAVING A HIGHLY-TUNED NERVOUS SYSTEM
CARING ABOUT EVERYONE AND EVERYTHING
HAVING STRONG EMOTIONAL REACTIONS
ALL OF THE FEELS ALL OF THE TIME

- I am “neurodivergent” (as opposed to “neurotypical”)
- My brain operates differently
- Important particularities
  - Hypersensitive to touch, washing hands frequently, does not touch h
    - Please don’t touch me or my belongings
  - Hypersensitive to noise
    - Please be quiet as we begin lectures, so I can concentrate
  - Hypersensitive to time pressures
    - Please be patient with me, given enough time I try to find the solutions that work for everyone
    - Don’t come to my office without an appointment
- When I am grounded, my goal is always to make you happy and fulfilled



Some non-tech interests

- Nature, going on hikes
- Art, doing art, seeing art
- Playing the **harpsichord** (and particularly Domenico Scarlatti)
- I love insects, particularly wasps and dragonflies (*they eat mosquitoes and other pests!*)



Sli.do ACCESS (reminder)

Audience participation with **Sli.do**

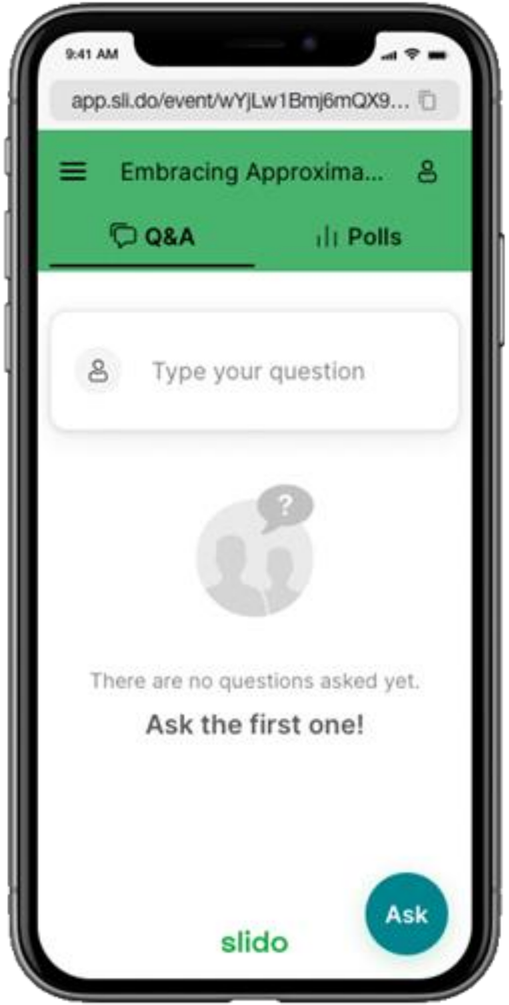
You can:

- **ask questions** and **upvote others' questions**
- **answer quizzes**
- choose to be anonymous (or not!)

Go to https://sli.do and enter the event code is #3017135

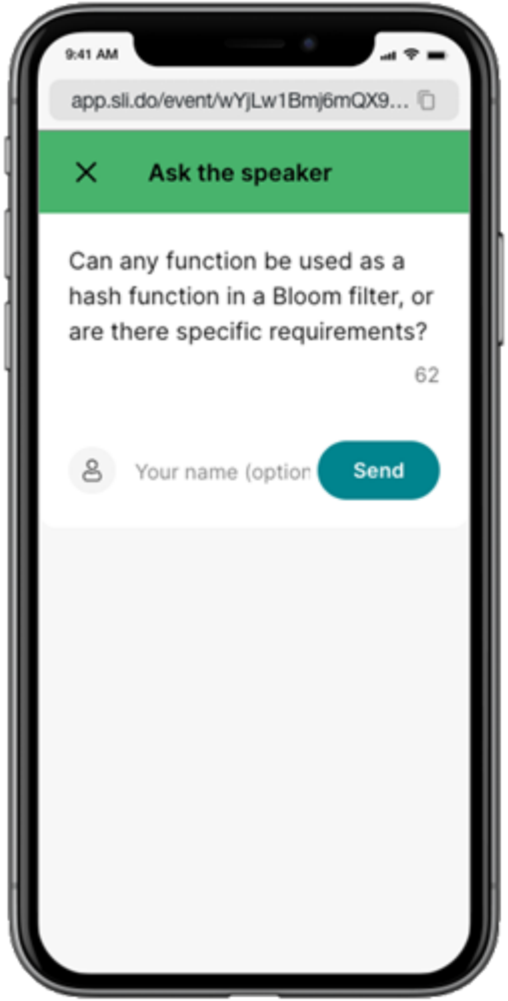
This should be a wonderful experience for you, please speak up!

slido.com with
#3017135

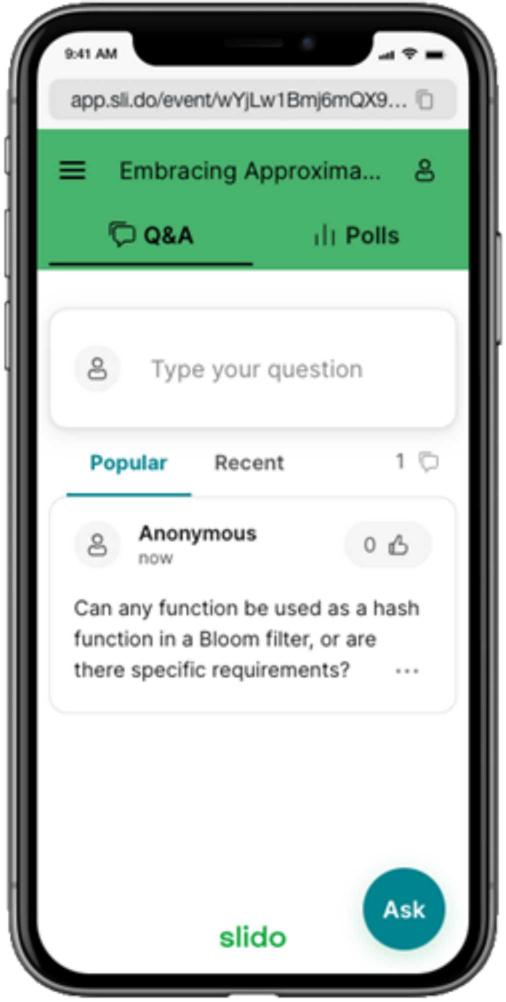


There are no questions asked yet.

Ask the first one!

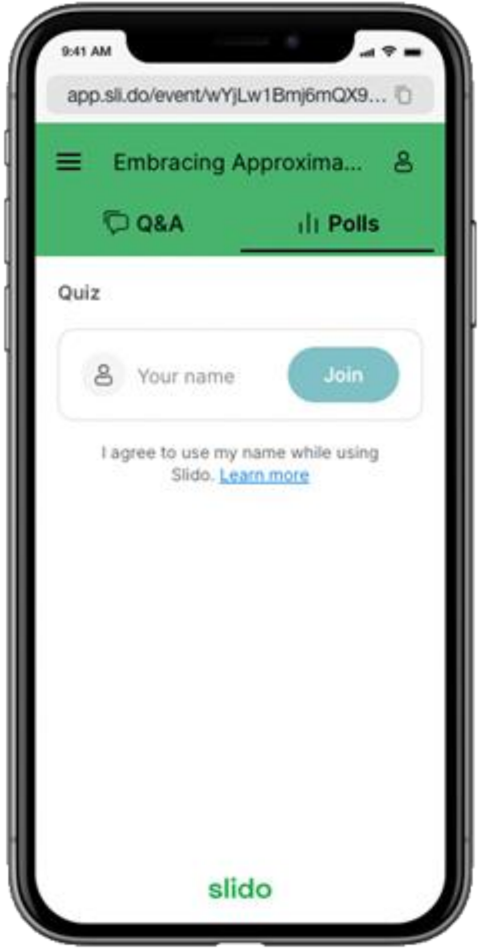


Can any function be used as a hash function in a Bloom filter, or are there specific requirements?



Can any function be used as a hash function in a Bloom filter, or are there specific requirements?

slido.com
with
#3017135



9:41 AM

app.sli.do/event/wYjLw1Bmj6mQX9...

Embracing Approxima...

Q&A

Polls

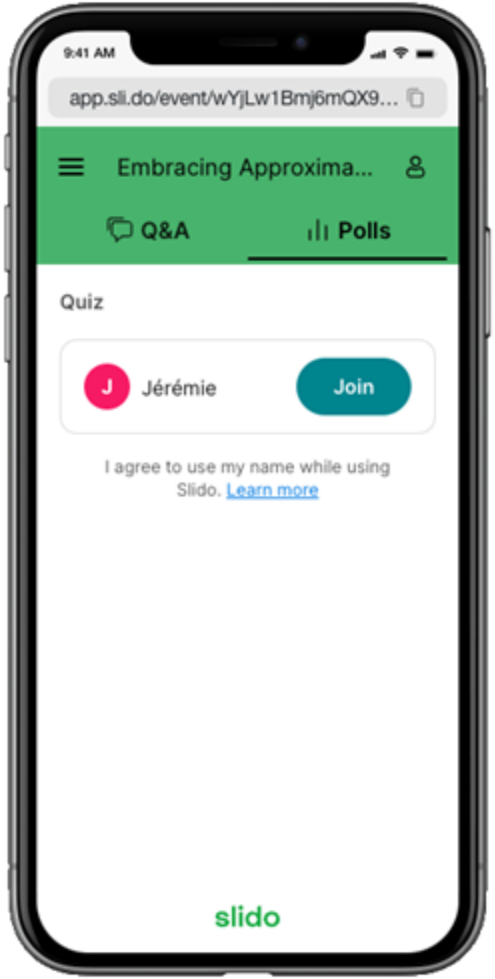
Quiz

Your name

Join

I agree to use my name while using Slido. Learn more

slido



9:41 AM

app.sli.do/event/wYjLw1Bmj6mQX9...

Embracing Approxima...

Q&A

Polls

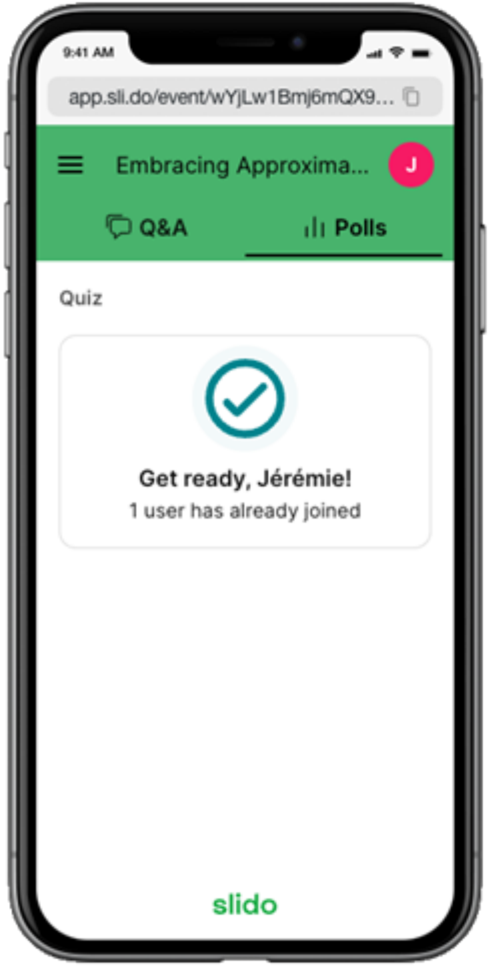
Quiz

J Jérémie

Join

I agree to use my name while using Slido. Learn more

slido



9:41 AM

app.sli.do/event/wYjLw1Bmj6mQX9...

Embracing Approxima... J

Q&A

Polls

Quiz

Get ready, Jérémie!

1 user has already joined

slido

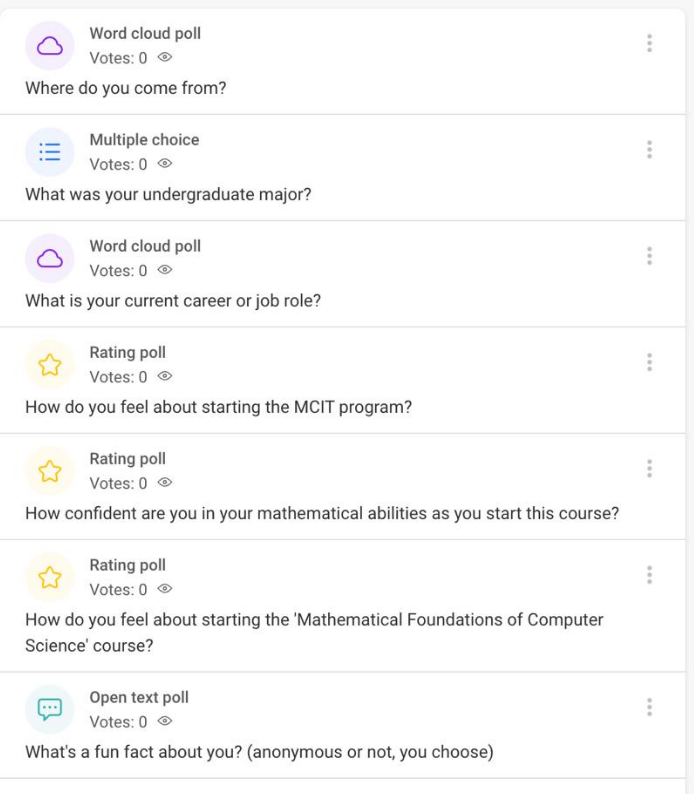
slido.com
with
#3017135

QUICK SURVEY ABOUT CLASS PEERS

Why do we have so many introductions

- It is important that you know we are here for you + can relate to us
- The value of a prestige campus education **is the netw**ork
- Learning is easier in collaboration
  - Find study partners
- It is hard to meet new people because of awkwardness
  - Fresh term is an opportunity
  - Once the freshness wears off, it becomes more awkward to enter cliques
- So we try to help you
- This afternoon’s recitation will also be another round of introductions

We will now do a quick survey to learn about the class



Word cloud poll
Votes: 0
Where do you come from?

Multiple choice
Votes: 0
What was your undergraduate major?

Word cloud poll
Votes: 0
What is your current career or job role?

Rating poll
Votes: 0
How do you feel about starting the MCIT program?

Rating poll
Votes: 0
How confident are you in your mathematical abilities as you start this course?

Rating poll
Votes: 0
How do you feel about starting the 'Mathematical Foundations of Computer Science' course?

Open text poll
Votes: 0
What's a fun fact about you? (anonymous or not, you choose)

slido.com with
#3017135

COURSE CONTENT

CIT 5920

- Mathematical Foundations of Computer Science
- Primary focus = discrete math
  - Discrete = that can be counted
- Does not cover
  - Any continuous math (calculus)
  - Linear algebra (matrices, vectors, etc.)
  - Optimization
  - Cryptography or any other difficult theory topics

Topics covered

- Basic combinatorics
  - Sets, functions, relations
  - Permutations and Combinations
- Discrete Probability and Expectations
- Elementary logic
- Introduction to mathematical proof methods
- Mathematical induction
- Introduction to graph theory

First we go UP the mountain (learning about objects);
Then we go DOWN the mountain (putting them into action in proofs)

PART 1
The Characters
INTRODUCING DISCRETE OBJECTS

- **Sets, functions, relations**
- **Permutations, combinations**
- **Probabilities**

PART 2
The Stories
HOW TO WRITE PROOFS

- **Elementary and propositional logic**
- **Elementary proofs**
- **Induction proofs**

PART 3
Getting Ready For The Wilderness
GRAPHS

- **Introduction to graph theory**

One of my favorite student course evaluations:

- “Jeremie Lumbroso--the best math Professor ever! I grew up taking math courses, but he is the only teacher who really made me learn something. He not only taught us math--he taught us the way to think about things, which is far more important than the concepts per se. He was always sincere. Some other math teachers like to impress: they learned things by heart themselves, and they talked as if they know everything since they were born. Jeremie is different. He paused to ponder over things, and he pointed out the ambiguity and the questions people may have. He is able to be vulnerable himself, which makes him super strong. I must say I got more than I expected from this course. Love you Jeremie! I'm sure 5920 will be popular since now because of you.”

Sets

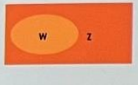
Definitions

A **set** is a collection of objects (such as numbers). Each member is called an **element** of the set. If A and B are sets, A is a **subset** of B if and only if every element of A is also an element of B. This is written $A \subseteq B$ which is read "A is a subset of B" or "B contains A."

If C and D are sets, C is a **proper subset** of D if and only if every element of C is an element of D, but there is at least one element of D that is not in C.

EX:
- The set N = {red, blue, green, yellow, purple, orange} has 6 elements.
- The set M = {red, green, yellow, orange} is a proper subset of N. Every element of M is an element of N, but there is at least one element in N that is not in M.

A **Venn diagram** is a visual representation that shows the relationship between sets.



EX: The set of whole numbers **W** = {0, 1, 2, 3, ...} is a proper subset of the set of integers **Z** = {... , –3, –2, –1, 0, 1, 2, 3, ...}.

Unions, Intersections & Complements

In mathematics, a **universal set**, U, is the set of all elements under consideration for a particular situation. The universal set could be the set of all integers, the set of all real numbers, the set of all positive integers, etc.

For a universal set U and sets A and B with $A \subseteq U$ and $B \subseteq U$, the following definitions hold.

- The **union** of A and B is the set of all elements x in U such that $x \in A$ or $x \in B$. The notation is $A \cup B$.
- The **intersection** of A and B is the set of all elements x in U such that $x \in A$ and $x \in B$. The notation is $A \cap B$.
- The **complement** of A is the set of all elements x in U such that $x \notin A$. The notation is A^c.

EX: Let $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $R = \{2, 4, 6, 8, 10\}$, and $S = \{1, 2, 3, 4, 5\}$.

$R \cup S = \{1, 2, 3, 4, 5, 6, 8, 10\}$

$R \cap S = \{2, 4\}$

$R^c = \{1, 3, 5, 7, 9\}$

$S^c = \{6, 7, 8, 9, 10\}$

Properties of Sets

For subsets A, B, and C of a universal set U, the following identities are true.

- **Associative Property**

  $(A \cap B) \cap C = A \cap (B \cap C)$

  $(A \cup B) \cup C = A \cup (B \cup C)$
- **Commutative Property**

  $A \cap B = B \cap A$

  $A \cup B = B \cup A$
- **Distributive Property**

  $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

  $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- **Complement of a Complement Property**

  $(A^c)^c = A$
- **De Morgan's Laws for Sets**

  $(A \cap B)^c = A^c \cup B^c$

  $(A \cup B)^c = A^c \cap B^c$

Ordered n-tuples & Cartesian Products

An **ordered n-tuple** is an arrangement of n elements of a set in which the order is important. The notation for an ordered n-tuple is $(x_1, x_2, \dots, x_n)$ in which x_1 is first, x_2 is second, and so on. An ordered 2-tuple is also called an **ordered pair** (x, y), and an ordered 3-tuple is called an **ordered triple** (x, y, z).

Two ordered n-tuples $(x_1, x_2, \dots, x_n)$ and $(y_1, y_2, \dots, y_n)$ are equal if and only if $x_1 = y_1, x_2 = y_2, \dots, x_n = y_n$.

For two sets A and B, the **Cartesian product** of A and B is the set of all ordered pairs (a, b), where $a \in A$ and $b \in B$. The notation for a Cartesian product is $A \times B$ which is read "A cross B."

EX: Given the sets A = {1, 2} and B = {3, 4, 5}, find A × B.

$A \times B = \{(1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5)\}$

The Empty Set

The **empty set** (or null set) is a unique set that contains no elements. The empty set is represented symbolically as $\varnothing$ or $\{\}$. There are several properties of sets that involve the empty set. For a universal set U and all sets A, the following properties are true:

- $A \cup \varnothing = A$
- $A \cap A^c = \varnothing$
- $A \cup A^c = U$
- $A \cap \varnothing = \varnothing$
- $U^c = \varnothing$
- $\varnothing^c = U$

Partitions of Sets

Two sets are **disjoint** if the sets have no elements in common.

A **partition** of a set A is a collection of nonempty sets $\{A_1, A_2, \dots, A_n\}$ such that $A = A_1 \cup A_2 \cup \dots \cup A_n$ and $A_1, A_2, \dots, A_n$ are mutually disjoint.

EX: Let A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. A partition of set A is $\{A_1, A_2, A_3\}$, where $A_1 = \{1, 2\}$, $A_2 = \{3, 4, 5, 6, 7\}$, and $A_3 = \{8, 9, 10\}$.

Counting

An experiment or process is **random** if it is not possible to predict the outcome with certainty for each trial. It is often useful to be able to count all of the possible outcomes of a random process or the subsets of a given set.

The Fundamental Counting Principle

The **Fundamental Counting Principle** says that if a process A has a possible outcomes, process B has b possible outcomes, process C has c possible outcomes, . . . , and process N has n possible outcomes, then the total number of ways all of the processes can occur is $a \cdot b \cdot c \cdot \dots \cdot n$.

EX: A deli offers 5 different kinds of bread, 8 different kinds of sandwiches, and 7 different side dishes. How many possible lunches are there if a customer selects a kind of bread, type of sandwich, and side dish at random?

There are $5 \cdot 8 \cdot 7 = 280$ possible lunches.

Combinations

A **combination** is a set of objects in which the order of the objects is not important.

EX: Let U = {Bryan, T

EX: There are 12 books on a shelf. Meredith selects 2 of the books at random to take on vacation. How many possible combinations of books are there?

$$n = 12, r = 2 \quad {}_{12}C_2 = \frac{12!}{(12-2)!2!} = \frac{12!}{10!2!} = 66$$

Permutations

A **permutation** is a set of objects in which the order of the objects is important. A set that contains n elements has $n!$ different permutations. This is because there are n possible outcomes for the first choice, $(n - 1)$ possible outcomes for the second choice, and so on.

EX: How many permutations are there of the elements of the set {red, blue, green}?

There are 3! = 6 permutations as shown below:

red, blue, green 🟥🟦🟩
red, green, blue 🟥🟩🟦
blue, red, green 🟦🟥🟩
blue, green, red 🟦🟩🟥

Counting & Probability

A **sample space** is the set of all possible outcomes for a random experiment or process. An **event** is a subset of a sample space of outcomes. The **probability** of an event is a numerical value that describes how likely the event is to occur. Probability ranges from 0 (impossible) to 1 (certain). It is calculated as the ratio of the number of favorable outcomes to the total number of possible outcomes in the sample space. The probability of an event E is denoted $P(E)$. The counting methods described earlier play an important role in calculating probability.

$$P(\text{event}) = \frac{\text{number of favorable outcomes}}{\text{total number of outcomes}}$$

EX: There are 6 sophomores, 8 juniors, and 10 seniors on a committee. A subcommittee of 6 students is chosen at random. What is the probability that the subcommittee contains 2 sophomores, 2 juniors, and 2 seniors?

Find the total number of outcomes (combinations)

Functions & Set Relations

Definitions

A **relation** R from set A to set B is any subset of the Cartesian product $A \times B$. For a given ordered pair (x, y) in $A \times B$, $x\,R\,y$ (read "x is related to y by R") if and only if $(x, y) \in R$.

EX: Let A = {1, 2, 3} and B = {4, 5}. Define the relation R to be any (x, y) in A × B such that $|y - x|$ is odd. Determine which ordered pairs are in R.

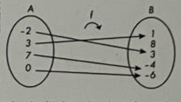
$A \times B = \{(1, 4), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5)\}$

Of these ordered pairs, only (1, 4), (2, 5), and (3, 4) satisfy the rule $|y - x|$ is odd. So, $R = \{(1, 4), (2, 5), (3, 4)\}$.

A **function** f from set A to set B is a relation between the elements of A and the elements of B such that each element of A is related to a unique element of B. The notation $f: A \to B$ is read "f is a function from A to B."

A is the **domain**, or the set of possible input values for the function. B is the **codomain**, or the set of possible output values for the function. The **range** of a function is the actual set of output values from the codomain that are paired with input values from the domain. A **mapping diagram** uses arrows to show how elements from the domain are paired with elements from the codomain.

EX: The mapping diagram shows a function f: A → B. Identify the domain, codomain, and range of the function. Then list the ordered pairs in f.



The domain is the set of possible input values, or {–2, 3, 7, 0}.

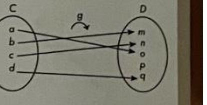
The codomain is the set of possible output values, or {1, 8, 3, –4, –6}.

The range is the set of output values that are paired with input values from the domain, or {1, 3, –6}. $f = \{(-2, 3), (3, 1), (7, -6), (0, -6)\}$

Injective, Surjective & Bijective Functions

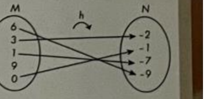
A function $f: A \to B$ is **injective**, or **one-to-one**, if and only if each element of the codomain is mapped by *at most* one element of the domain. In other words, if x_1 and x_2 and elements of the domain where $x_1 \neq x_2$, then $f(x_1) \neq f(x_2)$.

EX: The function g: C → D shown in the mapping diagram is injective because no two elements of the domain are mapped to the same element of the codomain.



A function $f: A \to B$ is **surjective**, or **onto**, if and only if for each element of the codomain there is at least one element in the domain that maps to it. In other words, if y_1 is any element of the codomain B, then there exists an element x_1 in A such that $f(x_1) = y_1$. For a surjective function, the range is equivalent to the codomain.

EX: The function h: M → N shown in the mapping diagram is surjective because each element of the codomain has an element of the domain that maps to it.

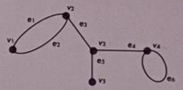


A function $f: A \to B$ that is both injective and surjective is a **bijective function**. Another way of saying that a function is bijective is to say that there is a **one-to-one correspondence** from A to B.

Graphs & Trees

Graphs

A **graph** is defined by a finite set of points, called **vertices**, and a finite set of line segments, called **edges**, that connect the vertices. Each edge is associated with either one or two vertices. An edge that is associated with just one vertex is called a **loop**. Two distinct edges that are associated with the same set of vertices are **parallel** edges. A **simple graph** is a graph that does not have any loops or parallel edges.



EX: Describe the edges and vertices of the graph.
- The vertices of the graph are $\{v_1, v_2, v_3, v_4, v_5\}$.
- The edges of the graph are $\{e_1, e_2, e_3, e_4, e_5, e_6\}$.
- Edges e_1 and e_2 are parallel because they are distinct edges that are both associated with vertices v_1 and v_2.
- Edge e_6 is a loop because it is associated with only one vertex, v_4.

Walks, Paths & Circuits

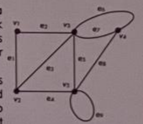
For a given graph with vertices v_1 and v_2, a **walk** from v_1 to v_2 is an alternating sequence of adjacent vertices and edges.

EX: In the preceding graph, one walk from v_1 to v_5 would be $v_1e_2v_2e_3v_3e_5v_5$.

A **path** from v_1 to v_2 is a walk that does not contain any repeated edges. A **simple path** is a path that does not contain any repeated vertices. The walk described in the previous example is a simple path because none of the edges or vertices in the walk repeat.

A **closed walk** is a walk that begins and ends at the same vertex. A **circuit** is a closed walk that does not contain a repeated edge. A **simple circuit** is a circuit with no repeated vertices except for the beginning and ending vertex.

EX: For this graph shown, describe each of the following walks as a path, a simple path, a circuit, or a simple circuit.

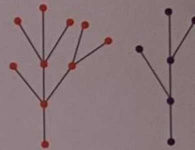


- $v_5e_8v_4e_6v_3e_3v_2e_1v_1$ is a simple path. It is a walk from v_5 to v_1 and has no repeated vertices or edges.
- $v_1e_2v_3e_6v_4e_7v_3e_3v_2e_1v_1$ is a circuit. It is a closed walk from v_1 to v_1 with no repeated edges, but it does have repeated vertices (v_3).
- $v_1e_2v_3e_6v_4e_7v_3e_5v_5$ is a path. It is a walk from v_1 to v_5 with no repeated edges, but it does have repeated vertices (v_3).
- $v_2e_3v_3e_5v_5e_4v_2$ is a simple circuit. It is a closed walk from v_2 to v_2 with no repeated edges or vertices other than the beginning and ending vertex.

Trees

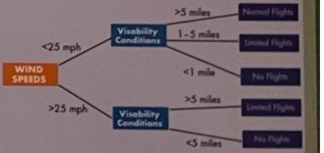
Two vertices of a graph, v_1 and v_2, are **connected** if and only if there is a walk from v_1 to v_2. A graph is connected if and only if every pair of vertices in the graph is connected. A **trivial circuit** is a circuit that consists of only a single vertex. A graph is a **tree** if and only if it is connected and has only trivial circuits.

EX: The graphs shown below are trees. Every pair of vertices is connected and the only circuits are trivial circuits.



Decision trees are used in a variety of situations to make decisions based on a number of factors.

EX: The decision tree below could be used to determine whether a hot air balloon company flies depending on the wind speeds and visibility conditions.



U.S. $6.95

What is the point? High-level answer

- To prepare you for
  - CIS 5020: Analysis of Algorithms
  - CIT 5960: Algorithms and Computation
  - Other algorithms or data structure courses
  - …

- To develop formal computational thinking reasoning

- **To be able to communicate, think in a formal way**

- This material will help you express yourself in the accurate and exacting way that is necessary when writing programs

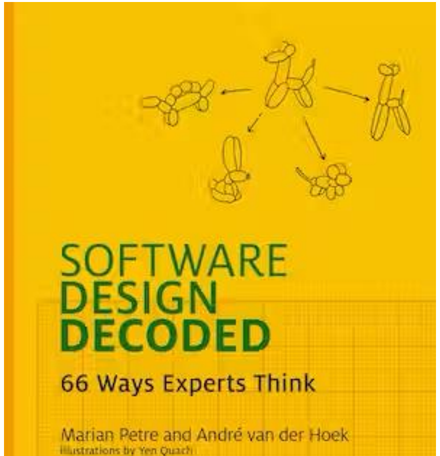
- To build your confidence and prepare you for independent math learning

Some linguistic ambiguities

1. "You can't miss it."
   1. Interpretation 1: It's so obvious that you won't miss it.
   2. Interpretation 2: You are not allowed to miss it.
2. "They are flying planes."
   1. Interpretation 1: They are piloting the planes.
   2. Interpretation 2: The objects that are flying are planes.
3. "Visiting relatives can be tiresome."
   1. Interpretation 1: The act of visiting relatives can be exhausting.
   2. Interpretation 2: The relatives who visit you can be tiresome.
4. "I saw the man with the telescope."
   1. Interpretation 1: I used a telescope to see the man.
   2. Interpretation 2: I saw a man who had a telescope with him.
5. "The chicken is ready to eat."
   1. Interpretation 1: The chicken is cooked and ready for consumption.
   2. Interpretation 2: The chicken is hungry and ready to eat something.

Notations as lens

- Being able to formally express yourself is helpful to approach unconventional problems
- (from Marian Petre and others, “Software Design Decoded”)

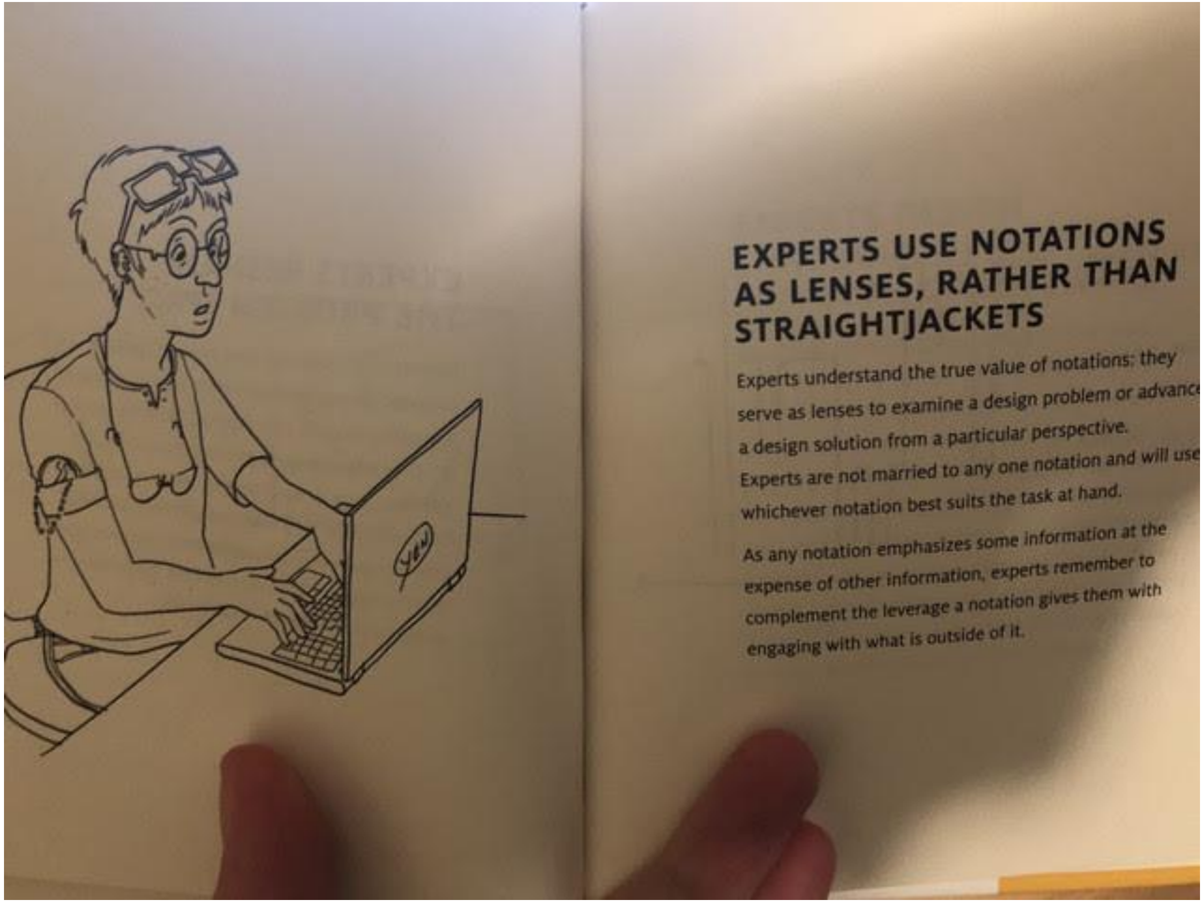


SOFTWARE DESIGN DECODED

66 Ways Experts Think

Marian Petre and André van der Hoek

Illustrations by Yen Quach



EXPERTS USE NOTATIONS AS LENSES, RATHER THAN STRAIGHTJACKETS

Experts understand the true value of notations: they serve as lenses to examine a design problem or advanc a design solution from a particular perspective. Experts are not married to any one notation and will use whichever notation best suits the task at hand.

As any notation emphasizes some information at the expense of other information, experts remember to complement the leverage a notation gives them with engaging with what is outside of it.

```
$ls -l | grep "Jan" | sort +2n | more
```

EXPERTS DESIGN ELEGANT ABSTRACTIONS

While all developers create abstractions, experts *design* them. A good abstraction makes evident what is important, both in what it does and how it does it. Through a single lens, it communicates the problem it solves and the machinery of its solution.

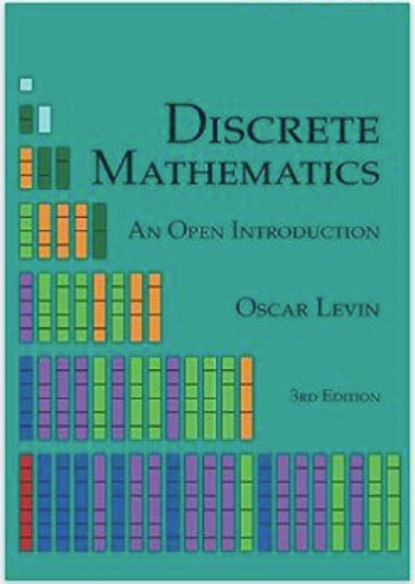
Experts are not satisfied with just any abstraction, they deliberately seek elegant abstractions through which complex structures can be introduced, understood, and referred to efficiently.

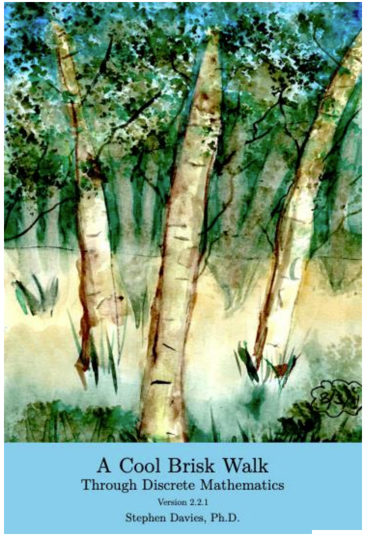
Lecture Notes & Textbooks

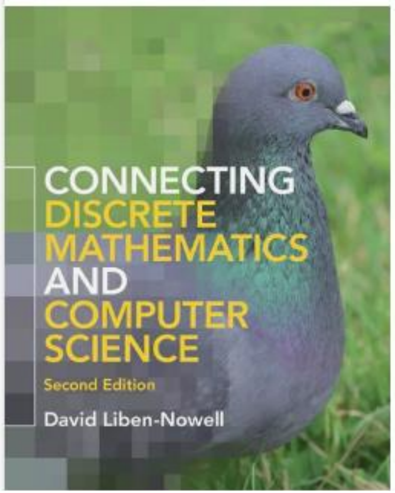
- We will be using online lecture notes (created by Professor Arvind Bhusnurmath)

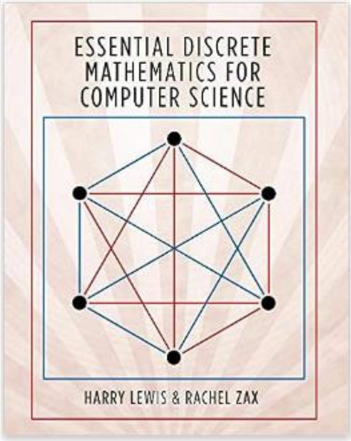
- There are many useful textbooks
- Most of them are free

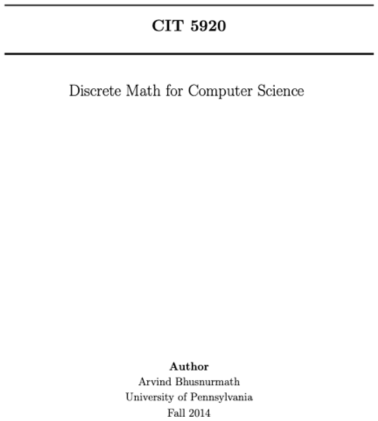
- DO NOT USE
  - Rosen
  - Lehman, Leighton and Meyer
  - **Anything that makes you feel dumb!**










CIT 5920

Discrete Math for Computer Science

Author
Arvind Bhusnurmath
University of Pennsylvania
Fall 2014

OVERVIEW OF THEMES

Math is needed for programming?

One of the most commonly asked questions in 592 is …

‘Where will I ever use this????’

Think of these topics as tools in your toolbox.

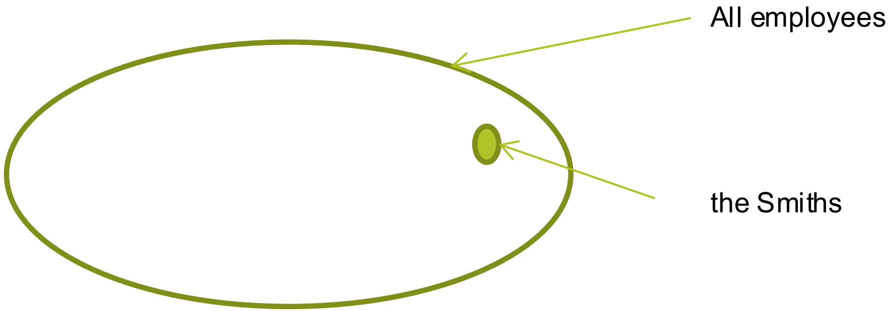
Tools in toolbox(??)

- But I thought the computer was the only tool I needed!
- Computers are dumb. They need humans to “think” for them.
- Math gives you the structured approach that is most directly associated with the way computer programs/algorithms are written
- It helps
  - Clean up problems from the real world
  - Focus on the important assumptions (sometimes change them)
  - Create *abstractions* to make it easier to solve problems
  - Notice and remember patterns
  - Think in a formal, organized, structured way

Sets

- Databases

SELECT EMP_ID, LAST_NAME FROM EMPLOYEE_TBL WHERE LAST_NAME = ‘Smith';



Logic

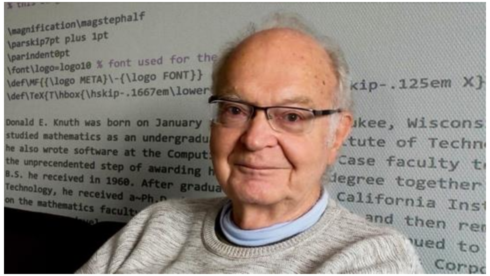
- Day to day logical reasoning does incorporate aspects of ‘formal logic’
    - All 592 students are in MCIT
    - All MCIT students are looking for new jobs
    - Therefore any student in this class needs help looking for a new job ….
- You can greatly simplify code if you can simplify logical statements.
- Imperative for digital circuit design.

Mathematical proof

- The most controversial topic taught in this course because …
- ‘I’ve never proven anything in the software industry’
  - Maybe you haven’t programmed airplanes/rockets/medical devices

BUT

- The ability to write a good proof is not too far removed from the ability to write a program with v few bugs
- Writing a proof requires translating/formalizing your thoughts in a non-ambiguous expression
- Very useful for an Algorithms course
- Therac bug, Ariane bug, Zune bug, etc.



“Beware of bugs in the above code; I have only proved it correct, not tried it.” – Donald Knuth

Discrete math will make you clearer

- Here many problems are abstracted by the notation
- Advantages
  - Generalization
  - Parameterization
  - Scalability
  - Conciseness
  - Abstraction
  - Reusability
  - Mathematical rigor
  - Fully specified

Problem: Model a rectangular grid of cells

Without Discrete Math

We have a grid with
3 rows and 5 columns

To build the grid:

1. Create an array
   with 3 elements

2. Each element is
   an array of 5 cells

With Discrete Math

Let n = number of rows,
m = number of columns

Grid $G(n, m) =$
$\{(i, j) \mid 0 \leq i < n,$
$0 \leq j < m\}$

For $n = 3$, $m = 5$:
$G(3, 5) = \{(i, j) \mid$
$0 \leq i < 3, 0 \leq j < 5\}$

Problem: Count combinations of k items from n total

Without Discrete Math

We have 10 items total
and want to choose 3.

To count combinations:

1. List all possible
   ways to choose 3
   items from 10
2. Count the total
   number of ways

With Discrete Math

Let n = total items,
k = items to choose

Number of combinations
$C(n, k) = n! / (k!(n-k)!)$

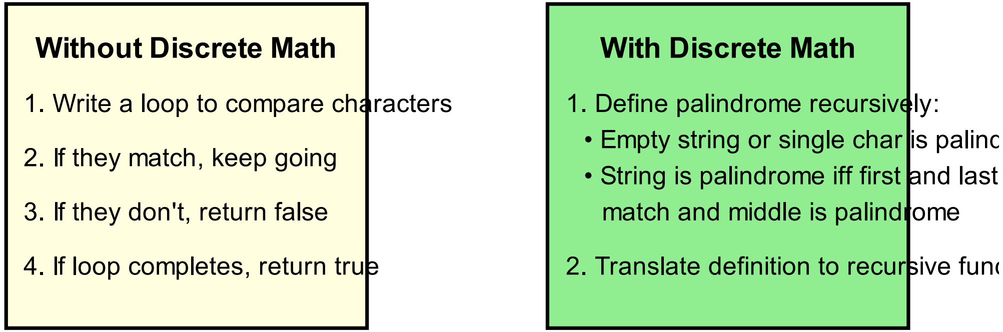
For $n = 10$, $k = 3$:
$C(10, 3) = 10! / (3!7!)$
$= 120$

So there are 120 ways
to choose 3 from 10.

Discrete math will make you faster

- Palindromes are words that can be read in reverse (madam, rotator, etc.)
- Here recursive definition is more efficient to test, because it can be broken down efficiently

Task: Write a function to check if a string is a palindrome



Without Discrete Math

1. Write a loop to compare characters
2. If they match, keep going
3. If they don't, return false
4. If loop completes, return true

With Discrete Math

1. Define palindrome recursively:
   - Empty string or single char is palinc
   - String is palindrome iff first and last match and middle is palindrome
2. Translate definition to recursive func

Discrete math will make you correct

- Prime numbers are numbers that cannot be divided by other numbers (like 2, 3, 5, 7, 11, etc.)

Problem: Determine if a number is prime

Without Discrete Math

1. Try dividing by first few numbers
2. If none divide evenly, assume prime
3. Write some code to test a few cases
4. If it works for those, ship it!
5. Oh no, a user found a bug! 😨
6. Try to patch the code
7. Repeat steps 5-6 indefinitely

With Discrete Math

1. Formally define prime numbers
2. Prove: p is prime iff p has no factors between 2 and √p
3. Use proof to write clear algorithm
4. Prove algorithm is correct
5. Implement algorithm, with confidence it's bug-free
6. Enjoy a robust solution! 😊

Counting/combinatorics usages

- How long is my program going to take?
  - Anyone can write inefficient code
  - A good programmer is able to analyze their program
  - = How long will battery last?
- Analysis of programs almost always begins with having some idea of the ‘number of operations’
- Larger the data, longer the time taken. But how does it scale?
- Computing probabilities often involves solving two counting problems.

Probability usages

- Where is probability used?
  - Machine learning
  - Las Vegas, Atlantic City, Monte Carlo, Macau
  - Making(and losing) millions and billions on Wall Street
  - Randomized algorithms
    - what is the ‘expected’ running time of quicksort

Mathematical induction

- Breaking up a problem into smaller problems
- Use the smaller problem solutions to solve the big problem
- When used in proofs = induction
- When used in programming = recursion!

Graph Theory

- Graph theory + probability + 2 PhD students + the internet = ….

Google™

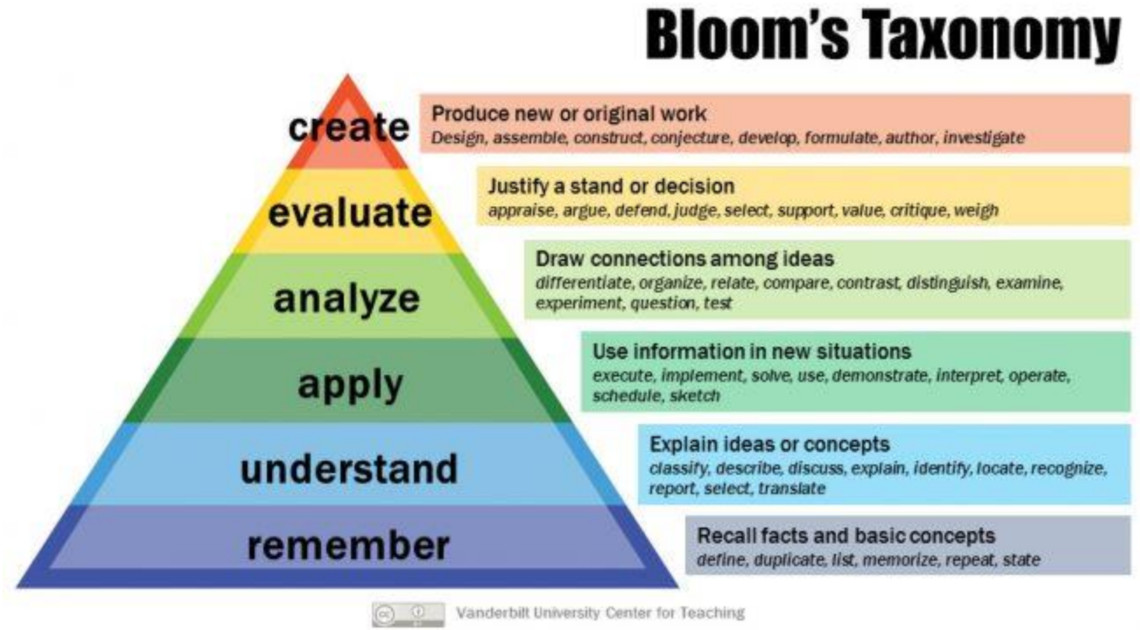
- Navigation applications
  - What is the shortest route from point A to point B?
- Any social networking site will have to use graph theory.
- Lots more …

ADMINISTRIVIA

Resources

- Class website: https://math4cs.university
- Course resources will be ready by the end of the week
- Class forum on Ed Discussion
  - Main method of communication, monitored by whole staff
  - Use private messages if necessary
- Homeworks and exams
  - Some live online questions on PrairieLearn
  - Some proof writing (in LaTeX on Overleaf, submitted as PDF)
  - Submission instructions to come later
- Office hours
  - Every TA + instructor will have office hours

Bloom’s taxonomy 1/3



Bloom’s Taxonomy

create
Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate

evaluate
Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh

analyze
Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test

apply
Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch

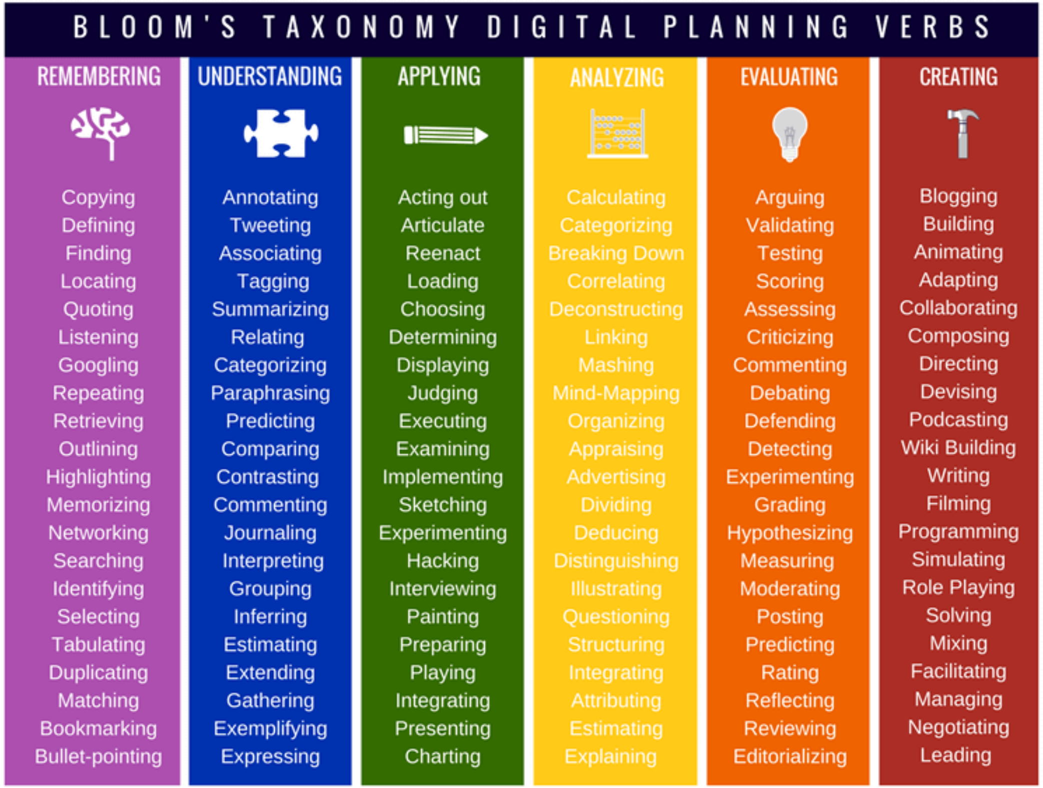
understand
Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate

remember
Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

Vanderbilt University Center for Teaching

- For each topic, learning involves going up an increasingly more difficult pyramid
- Each level builds on the ones below

Bloom’s taxonomy 2/3



BLOOM'S TAXONOMY DIGITAL PLANNING VERBS

REMEMBERING	UNDERSTANDING	APPLYING	ANALYZING	EVALUATING	CREATING
Copying	Annotating	Acting out	Calculating	Arguing	Blogging
Defining	Tweeting	Articulate	Categorizing	Validating	Building
Finding	Associating	Reenact	Breaking Down	Testing	Animating
Locating	Tagging	Loading	Correlating	Scoring	Adapting
Quoting	Summarizing	Choosing	Deconstructing	Assessing	Collaborating
Listening	Relating	Determining	Linking	Criticizing	Composing
Googling	Categorizing	Displaying	Mashing	Commenting	Directing
Repeating	Paraphrasing	Judging	Mind-Mapping	Debating	Devising
Retrieving	Predicting	Executing	Organizing	Defending	Podcasting
Outlining	Comparing	Examining	Appraising	Detecting	Wiki Building
Highlighting	Contrasting	Implementing	Advertising	Experimenting	Writing
Memorizing	Commenting	Sketching	Dividing	Grading	Filming
Networking	Journaling	Experimenting	Deducing	Hypothesizing	Programming
Searching	Interpreting	Hacking	Distinguishing	Measuring	Simulating
Identifying	Grouping	Interviewing	Illustrating	Moderating	Role Playing
Selecting	Inferring	Painting	Questioning	Posting	Solving
Tabulating	Estimating	Preparing	Structuring	Predicting	Mixing
Duplicating	Extending	Playing	Integrating	Rating	Facilitating
Matching	Gathering	Integrating	Attributing	Reflecting	Managing
Bookmarking	Exemplifying	Presenting	Estimating	Reviewing	Negotiating
Bullet-pointing	Expressing	Charting	Explaining	Editorializing	Leading

- As we process a concept, we move from the “Remembering” to eventually the “Creating”

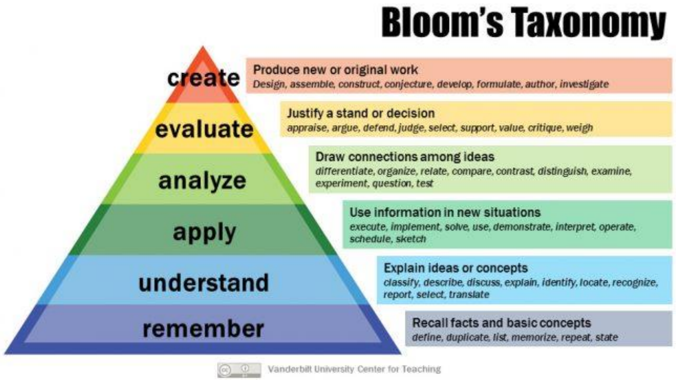
Bloom’s Taxonomy 3/3

- **Lectures**
  - **The lecture contains the primary introduction of content**
  - **SLI.DO POLLS are meant to help you remember and understand**
    - When you answer the questions, you are checking that you heard me correctly
    - You will then review these questions when studying for an exam

- Homeworks
  - PRAIRIE LEARN questions help you apply the course material
    - You will have **infinite attempts on all questions and will be able to practice them throughout the semester**
  - The WRITTEN QUESTIONS are designed to help you analyze and evaluate

- Exams
  - The EXAMS are meant to evaluate that you
    - Remember
    - Understand
    - Can apply
    - Analyze

    Based on the course materials
  - Easier but timed, and synchronous



Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate

Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh

Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test

Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch

Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate

Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

Vanderbilt University Center for Teaching

“No Time Pressures” Policy

- The deadlines we provide in the course should — as much as possible — be respected
- BUT
  - The goal of the deadlines it to keep us all on pace and to help logistics
  - It is NOT to punish you, or to set a normative pace

- Submit the extension request form, you can assume (unless you hear otherwise) that your extension is approved:
  - **Must be requested before the deadline**
  - How many days do you think you need? (You can always ask for more.)
  - Why do you need the extension? (optional)

- Other side of the coin: Work submitted late may be graded very slowly

Grading

You final grade will be based on

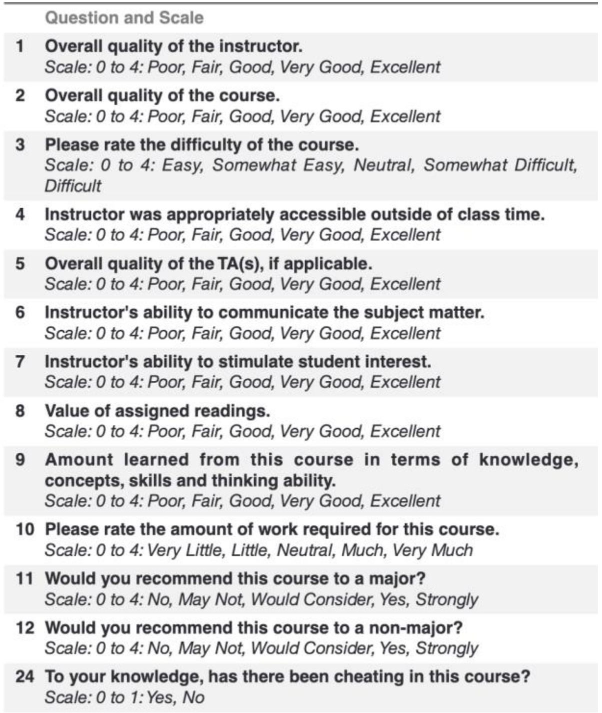
- Weekly HWs - 45%
- Exams - 45%
- Academic honesty and professionalism – 5%
- Engagement and participation – 5%

- There will be 3 exams. One of them will be during the final exam period but will not be any lengthier or ”scarier”.

17, 15, 13

- Your best exam will be worth 17%, second best 15%, and worst exam is worth 13% of your grade.
- Trying to strike some balance between rewarding consistency and also not penalizing one bad day too much.
- If your HW performance disadvantages you, we will weight exams higher.
- **Bottom line: We are going to try to make grading as stress-free as possible.**

Course evaluations



	Question and Scale
1	**Overall quality of the instructor.** *Scale: 0 to 4: Poor, Fair, Good, Very Good, Excellent*
2	**Overall quality of the course.** *Scale: 0 to 4: Poor, Fair, Good, Very Good, Excellent*
3	**Please rate the difficulty of the course.** *Scale: 0 to 4: Easy, Somewhat Easy, Neutral, Somewhat Difficult, Difficult*
4	**Instructor was appropriately accessible outside of class time.** *Scale: 0 to 4: Poor, Fair, Good, Very Good, Excellent*
5	**Overall quality of the TA(s), if applicable.** *Scale: 0 to 4: Poor, Fair, Good, Very Good, Excellent*
6	**Instructor's ability to communicate the subject matter.** *Scale: 0 to 4: Poor, Fair, Good, Very Good, Excellent*
7	**Instructor's ability to stimulate student interest.** *Scale: 0 to 4: Poor, Fair, Good, Very Good, Excellent*
8	**Value of assigned readings.** *Scale: 0 to 4: Poor, Fair, Good, Very Good, Excellent*
9	**Amount learned from this course in terms of knowledge, concepts, skills and thinking ability.** *Scale: 0 to 4: Poor, Fair, Good, Very Good, Excellent*
10	**Please rate the amount of work required for this course.** *Scale: 0 to 4: Very Little, Little, Neutral, Much, Very Much*
11	**Would you recommend this course to a major?** *Scale: 0 to 4: No, May Not, Would Consider, Yes, Strongly*
12	**Would you recommend this course to a non-major?** *Scale: 0 to 4: No, May Not, Would Consider, Yes, Strongly*
24	**To your knowledge, has there been cheating in this course?** *Scale: 0 to 1: Yes, No*

- You get evaluated with exams + course grade
- I get evaluated with the evaluations you submit
- I take the evaluations very seriously
- Please take it seriously too and remember that despite your anonymity your comments have an impact
- Please provide feedback during the semester to give me a chance to address any issues you may have
- Feedback is LOVE

Feedback is LOVE

I may not always agree, and I am strongly opinionated, but I always listen, and often find a way to give you satisfaction:

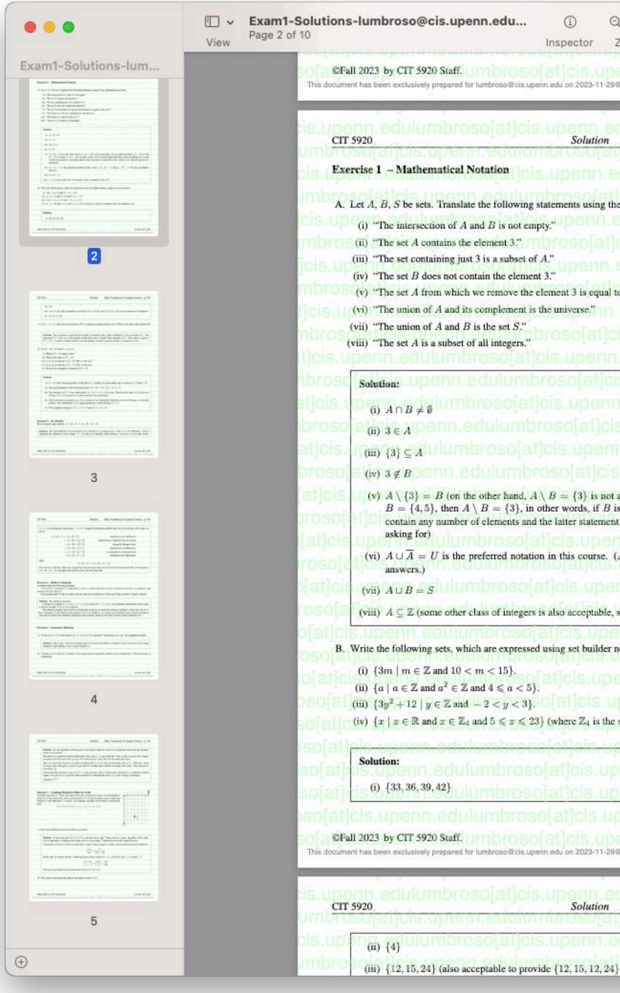
- “more than willing to take students' opinions”
- “incorporates student feedback into materials”
- “took in student input where he could”
- “Very receptive to student feedback (the most receptive lecturer I have ever seen).”
- “always seek ways to [improve].”
- “cared about what students had to say”
- “actively took in feedback from the students”

Academic honesty

- Academic dishonesty in this class is a waste of everyone’s time
  - Copying, cheating, abusing LLMs
- This course is
  - Your money
  - Your time
  - Your opportunity cost (the things you could be doing instead)
- You don’t have to be here: If you want to be waived out of this course, please let me know
- A university diploma is more than a piece of paper
- In my experience, MCIT students are among the most mature I have ever had
- Do. Not. Share. Course. Solutions.

Do. Not. Share. Course. Solutions.

- For new classes
  - 100% of the time is just creating + running course material
- For classes that have already run
  - 60% of the time is running the class
  - 40% can be spent improving materials, refining explanations, etc.
- When solutions leak, instructors recreate materials
- This is a HUGE waste of time and it penalizes everyone
- Furthermore, anybody who shares solution with you is not your friend, they are undermining your education



©Fall 2023 by CIT 5920 Staff.

This document has been exclusively prepared for lumbroso@cis.upenn.edu on 2023-11-29

CIT 5920 *Solution*

Exercise 1 – Mathematical Notation

A. Let A, B, S be sets. Translate the following statements using the

(i) "The intersection of A and B is not empty."

(ii) "The set A contains the element 3."

(iii) "The set containing just 3 is a subset of A."

(iv) "The set B does not contain the element 3."

(v) "The set A from which we remove the element 3 is equal to

(vi) "The union of A and its complement is the universe."

(vii) "The union of A and B is the set S."

(viii) "The set A is a subset of all integers."

Solution:

(i) $A \cap B \neq \emptyset$

(ii) $3 \in A$

(iii) $\{3\} \subseteq A$

(iv) $3 \notin B$

(v) $A \setminus \{3\} = B$ (on the other hand, $A \setminus B = \{3\}$ is not a $B = \{4, 5\}$, then $A \setminus B = \{3\}$, in other words, if B is contain any number of elements and the latter statement asking for)

(vi) $A \cup \overline{A} = U$ is the preferred notation in this course. (answers.)

(vii) $A \cup B = S$

(viii) $A \subseteq \mathbb{Z}$ (some other class of integers is also acceptable,

B. Write the following sets, which are expressed using set builder n

(i) $\{3m \mid m \in \mathbb{Z} \text{ and } 10 < m < 15\}$.

(ii) $\{a \mid a \in \mathbb{Z} \text{ and } a^2 \in \mathbb{Z} \text{ and } 4 \leqslant a < 5\}$.

(iii) $\{3y^2 + 12 \mid y \in \mathbb{Z} \text{ and } -2 < y < 3\}$.

(iv) $\{x \mid x \in \mathbb{R} \text{ and } x \in \mathbb{Z}_4 \text{ and } 5 \leqslant x \leqslant 23\}$ (where $\mathbb{Z}_4$ is the

Solution:

(i) $\{33, 36, 39, 42\}$

©Fall 2023 by CIT 5920 Staff.

This document has been exclusively prepared for lumbroso@cis.upenn.edu on 2023-11-29

CIT 5920 *Solution*

(ii) $\{4\}$

(iii) $\{12, 15, 24\}$ (also acceptable to provide $\{12, 15, 12, 24\}$

Professionalism

- Everyone starts off with 5 points
- Be nice to your fellow students
- Do not harass a TA into providing you answers
  - We cannot check your answer before you submit it

- Learn how to say you don’t understand something without being embarrassed

Engagement and participation

- I want you to engage with the course material
- You should be planning to spend
  - AT LEAST 3-4 hours outside of class STUDYING (reading textbook, going over notes, doing practice exercises)
  - AT LEAST 4-6 hours for homework
- You should plan to interact with us (office hours, Slido, or forum)
- Engagement opportunities
  - Upload photo on Canvas, Ed Discussion, on Gmail
  - Respond to other people's questions
  - Privately report issues with the materials
  - Help make this a great experience for everyone

LaTeX

- Most popular tool for writing math
- http://en.wikibooks.org/wiki/LaTeX
- overleaf.com
  - Penn provides all students and faculty with a free account
  - Login with your @upenn.edu address
- Recitation 1 will have a demo to show you how to use it

HW submission

- **Part of it will be on PrairieLearn (online)**
- **Proofs will be in LaTeX**
- **Only in pdf form and all on canvas**
- **NO IMAGES OF HANDWRITTEN SOLUTION**
- **Homeworks will generally be due on Monday 11:59PM**

Recitation

TAs will run the recitation.

You will be doing practice problems in the recitations.

They will let you know more about the format next Friday.

SPECIAL RECITATION THIS FRIDAY

- **Introductions among peers (speed networking)**
- **Get started with LaTeX on Overleaf**

Collaboration policy

- You are allowed to collaborate on HW with 1 other person.
- Write the name of your collaborator at the top of your HW.
- In the end you have to write your own solution. NO COPY PASTE ALLOWED.
- Individual submission
- You and your collaborator can and usually will get different scores on HW.
- You are allowed to change collaborators.

AI policy

- Unrestricted Use of Foundation Models:
  - Allowed to enhance assignment quality.
- Critical Approach to AI Outputs:
  - Quote: "Forces me to think critically about every sentence" - Arvind Narayan, April 9, 2023.
  - Beware of potential inaccuracies and biases. AI is most useful to REFLOW + COMBINE text.
- AI as an Assistant:
  - Students must vet and understand all AI-generated content.
  - Think of AI as a tool, not a replacement.
- Acknowledgment Requirement:
  - All AI-influenced submissions must cite foundation model use.
- Emphasis on Understanding:
  - Students expected to explain and defend their work.
  - Focus on critical reasoning over memorization.
- Goal:
  - Leverage AI for higher quality work while maintaining academic integrity.

A suggested approach to collaboration

1. Identify collaborator
    1. Pick someone who has roughly the same skill set.
    2. Someone with a similar schedule (*see study habits survey in recitation 1*)
2. Do it yourself first
    1. Come to office hours if stuck (or Piazza post)
3. Meet and discuss with your collaborator
    1. Come to office hours again if needed.
4. Write the answers by yourself. DO NOT copy from your collaborator.

THANK
YOU