Before we start

If you or someone you know is suffering from food and/or housing insecurities there are UCSD resources here to help:

Basic Needs Office: https://basicneeds.ucsd.edu/

Triton Food Pantry (in the old Student Center) is free and anonymous, and includes produce:

https://www.facebook.com/tritonfoodpantry/

Mutual Aid UCSD: https://mutualaiducsd.wordpress.com/

If you find yourself in an uncomfortable situation, ask for help. We are committed to upholding University policies regarding nondiscrimination, sexual violence and sexual harassment.

Counseling and Psychological Services (CAPS) at 858 5343755 or http://caps.ucsd.edu

OPHD at (858) 534-8298, ophd@ucsd.edu , http://ophd.ucsd.edu. CARE at Sexual Assault Resource Center at 858 5345793 sarc@ucsd.edu http://care.ucsd.edu

Pandemic resilient instruction

Spring 2022 feels like the first quarter post-pandemic. We were all kind of blindsided with Omicron last quarter and so we should all be cautious about what we expect to happen. We will keep a close eye on university policies and will inform the class whenever there is a change to our class policy.

I will be giving lectures in-person. I will also offer remote support throughout the quarter. Throughout the quarter, the "in-person" lectures will be live-streamed through zoom and recorded via "podcast.ucsd.edu". The exams will be offered "in-person" or "remote" and each student can choose which type of exam they prefer. This way we can be ready for whatever lies ahead and also give students the option of how they want to interact with the class.

First and foremost is the health and safety of everyone. Please do not come to class if you are sick or even think you might be sick. It is likely that the university will be requiring masks and "symptom screeners" and/or "covid tests". We expect all students to follow the rules. With all of this in mind, we encourage all students to come to class when they can, but will also provide as much of the class materials as we can in a remotely viewable format. The lectures are designed to engage students in real time with opportunities for questions and discussions between instructor and students and also between students and other students. We will also have some ways for students who participate remotely to engage in discussions with the instructors and other students, but cannot guarantee the full experience for remote students.

(Personal note: Last quarter (Winter 2022), we started online and returned to campus mid-quarter. After that, I ran my classes in a similar way. I am hopeful to expect that Spring quarter will be fully in person but I am a bit wary. For that reason, I have decided to continue offering the remote option for Spring 2022 mainly due to the fact that I do not know what the future will hold. There are many things about teaching remotely that I found to be great and I will try to incorporate some lessons learned from my remote teaching experience. I am excited and hopeful to teach a quarter fully "in-person". I will try my best to bring a classroom experience that is the best of both worlds. -Miles Jones)

Welcome to CSE 20: Discrete Math for Computer Science in Fall 2021!

Themes and applications for CSE 20

- **Technical skepticism**: Know, select and apply appropriate computing knowledge and problem-solving techniques. Reason about computation and systems. Use mathematical techniques to solve problems. Determine appropriate conceptual tools to apply to new situations. Know when tools do not apply and try different approaches. Critically analyze and evaluate candidate solutions.

- **Multiple representations**: Understand, guide, shape impact of computing on society/the world. Connect the role of Theory CS classes to other applications (in undergraduate CS curriculum and beyond). Model problems using appropriate mathematical concepts. Clearly and unambiguously communicate computational ideas using appropriate formalism. Translate across levels of abstraction.

Applications: Numbers (how to represent them and use them in Computer Science), Recommendation systems and their roots in machine learning (with applications like Netflix), "Under the hood" of computers (circuits, pixel color representation, data structures), Codes and information (secret message sharing and error correction), Bioinformatics algorithms and genomics (DNA and RNA).

Introductions

Class website: There is a link to a class website from my homepage: https://cseweb.ucsd.edu/~mej016/

I am also using Canvas as a website.

(I am transitioning from Canvas to a new resource that the department has been developing. It has been initiated by Mia Minnes and Joe Politz with help from Aiko Coanaya and Akanksha Pandey.)

Pro-tip: you can use MATH109 to replace CSE20 for prerequisites and other requirements.

Instructor: Miles Jones

Our team: Five TAs and 8 tutors + all of you

On a typical week: **TuTh** Lectures + review quizzes, **W** HW due, **M** Discussion, office hours, Piazza. Tests and Midterms see the content calendar.

Tuesday March 29

What data should we encode about each Netflix account holder to help us make effective recommendations?

In machine learning, clustering can be used to group similar data for prediction and recommendation. For example, each Netflix user's viewing history can be represented as a n-tuple indicating their preferences about movies in the database, where n is the number of movies in the database. People with similar tastes in movies can then be clustered to provide recommendations of movies for one another. Mathematically, clustering is based on a notion of distance between pairs of n-tuples.

In the table below, each row represents a user's ratings of movies: ✓ (check) indicates the person liked the movie, ✗ (x) that they didn't, and ● (dot) that they didn't rate it one way or another (neutral rating or didn't watch). Can encode these ratings numerically with 1 for ✓ (check), -1 for ✗ (x), and 0 for ● (dot).

Person	Fyre	Frozen II	Picard	Ratings written as a 3-tuple
P_1	✗	●	✓	
P_2	✓	✓	✗	
P_3	✓	✓	✓	
P_4	●	✗	✓	

Conclusion: Modeling involves choosing data types to represent and organize data

Discrete Math vs Continuous Math

Continuous Math

 image reference

Continuous math is the study of the *real number line*
What are some subjects that would be considered as continuous math?

Discrete Math

 image reference

Discrete math deals with distinct objects (like the integers.)
What are some subjects that would be considered as discrete math?

Math and computer science.

What are some ways that math is used in computer science?

Notation and prerequisites

Term	Notation Example(s)	We say in English ...
sequence	$x_1, \ldots, x_n$	A sequence x_1 to x_n
summation	$\sum_{i=1}^{n} x_i$ or $\displaystyle\sum_{i=1}^{n} x_i$	The sum of the terms of the sequence x_1 to x_n
all reals	$\mathbb{R}$	The (set of all) real numbers (numbers on the number line)
all integers	$\mathbb{Z}$	The (set of all) integers (whole numbers including negatives, zero, and positives)
all positive integers	$\mathbb{Z}^+$	The (set of all) strictly positive integers
all natural numbers	$\mathbb{N}$ or $\mathbb{Z}^{\geq 0}$	The (set of all) natural numbers. **Note**: we use the convention that 0 is a natural number.
piecewise rule definition	$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$	Define f of x to be x when x is nonnegative and to be $-x$ when x is negative
function application	$f(7)$ $f(z)$ $f(g(z))$	f of 7 **or** f applied to 7 **or** the image of 7 under f f of z **or** f applied to z **or** the image of z under f f of g of z **or** f applied to the result of g applied to z
absolute value	$|-3|$	The absolute value of -3
square root	$\sqrt{9}$	The non-negative square root of 9

Mathematical object and data types

A mathematical object is any abstract object arising in mathematics.

What are some examples of mathematical objects?

Computers encode mathematical objects into data types. What are some examples of data types?

Data Types: sets, n-tuples, and strings

Term	Examples: (add additional examples from class)
set unordered collection of elements *repetition doesn't matter* *Equal sets agree on membership of all elements*	$7 \in \{43, 7, 9\}$ $2 \notin \{43, 7, 9\}$
n-tuple (also called an n-**dimensional vector**) ordered sequence of entries with n "slots" $(n > 0)$ *repetition matters, fixed length* *Equal n-tuples have corresponding entries equal*	
string ordered finite sequence of elements each from specified set *repetition matters, arbitrary finite length* *Equal strings have same length and corresponding characters equal*	

Special cases:

When $n = 2$, the 2-tuple is called an **ordered pair**.

A string of length 0 is called the **empty string** and is denoted λ.

A set with no elements is called the **empty set** and is denoted $\{\}$ or $\emptyset$.

Sets

A set is an unordered collection of *elements.*

Recall that the empty set (denoted $\emptyset$ or $\{\ \}$) is the set without any elements.

To define sets:

To define a set using **roster method**, explicitly list its elements using *curly braces.* That is, start with $\{$ then list elements of the set separated by commas and close with $\}$.

To define a set using **set builder definition**, either form "The set of all x from the universe U such that x is ..." by writing
$$\{x \in U \mid ...x...\}$$
or form "the collection of all outputs of some operation when the input ranges over the universe U" by writing
$$\{...x... \mid x \in U\}$$

We use the symbol $\in$ as "is an element of" to indicate membership in a set.

Example sets: For each of the following, identify whether it's defined using the roster method or set builder notation and give an example element.

$\{-1, 1\}$

$\{0, 0\}$

$\{-1, 0, 1\}$

$\{(x, x, x) \mid x \in \{-1, 0, 1\}\}$

$\{\}$

$\{x \in \mathbb{Z} \mid x \geq 0\}$

$\{x \in \mathbb{Z} \mid x > 0\}$

$\{\texttt{A}, \texttt{C}, \texttt{U}, \texttt{G}\}$

$\{\texttt{AUG}, \texttt{UAG}, \texttt{UGA}, \texttt{UAA}\}$

n-tuples

An n-tuple is a sequences of n *entries*.

To define n-tuples

To define n-tuples, list the entries separated by commas and enclosed by parentheses.

Example n-tuples: For each of the following n-tuple, come up with where it might come from. For example, $(88, 79, 100, 90, 68, 81, 90)$ may be the grades a particular student will get for homeworks 1-7.

$(1, 2.22, -0.8)$

$(-1, -1, 0, 1, 0, -1, 0, 0)$

$(CSE20, CSE21, CSE101, CSE105)$

$(MILES, JONES, A000000, MUIR, 91.86, A-)$

strings (over a finite alphabet)

A string over a finite alphabet is a finite sequence of *characters* or *symbols*.

Recall that the *empty string*, (usually denoted λ) is the string without any characters.

The characters you are allowed to use come from a finite set called an *alphabet*.

To define strings:

Usually, a string is given as a list of characters without any punctuation. In some cases, we will use the symbol "$\circ$" as a delimiter. For example, if the alphabet is $\{A, C, U, G\}$, then consider the string `AAAUA`. It could also be written as $A \circ A \circ A \circ U \circ A$.

string concatenation

String concatenation of strings x and y, denoted xy or $x \circ y$ is the result of "gluing together" the strings x first then y. For example: $\texttt{AUAA} \circ \texttt{GGC} = \texttt{AUAAGGC}$.

Pro-tip: the meaning of writing one element next to another like xy depends on the data-types of x and y. When x and y are strings, the convention is that xy is the result of string concatenation. When x and y are numbers, the convention is that xy is the result of multiplication. This is (one of the many reasons) why is it very important to declare the data-type of variables before we use them.

Example strings: For each example, come up with an appropriate alphabet:

$MATHEMATICS$

$A00000000$

2598960

`CCACUGCCA`

λ

$9500\ Gilman\ Dr.$

$111000101011010001111111011111$

$mej016@eng.ucsd.edu$

$F96A4B0B$