CSCI 338

Welcome to Computer Theory!

Instructor details

- Adiesha Liyanage
- Email: a.liyanaralalage@montana.edu
- Office hours:
  - Monday: 9.05-10.00 A.M
  - Tuesday: 10.00-11.00 A.M
  - Friday: 9.05-10.00 A.M
- Office: Barnard 349 (right next to conference room)

TA details

- **Name:** Caleb Eardley
- **Day(s):** Thursday
- **Time:** 10AM-12PM
- **Location:** Student Success Center
- **Email: caleb.eardley@student.montana.edu**

Let’s look at several problems

True or False?

- Finding shortest path in a distance-weighted graph can be done in polynomial time

True or False?

- Finding shortest path in a distance-weighted graph can be done in polynomial time

True

True or False?

Finding the longest simple path (no repeated vertices) in a given undirected graph can be done in polynomial time.

True or False?

Finding the longest simple path (no repeated vertices) in a given undirected graph can be done in polynomial time.

Unknown

True or False?

- Can you build a program that will answer the question: "Will this input code terminate eventually on a given input?"

True or False?

- Can you build a program that will answer the question: "Will this input code terminate eventually on a given input?"

Impossible regardless of the computer (Unsolvable)

Mathematical model

- A rigorous mathematical formulation of reality.
- Used to make predictions.

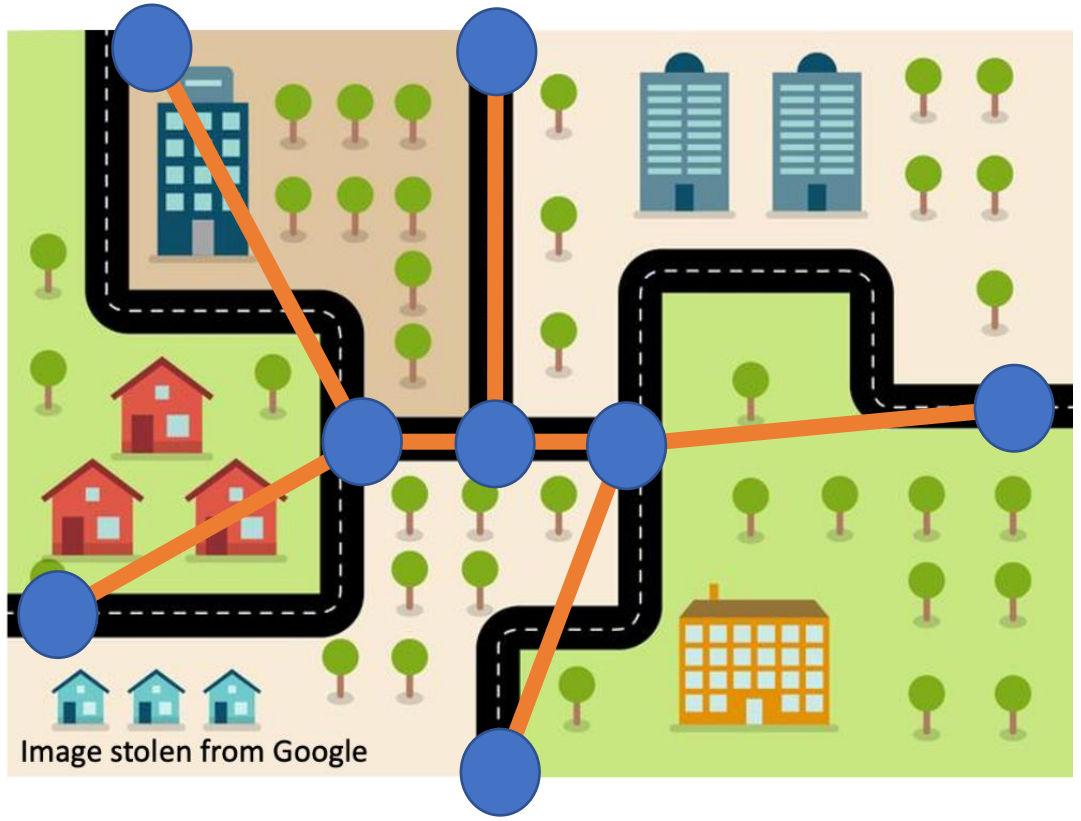
Can we represent a road network using a mathematical model?

Can we represent a road network using a mathematical model?

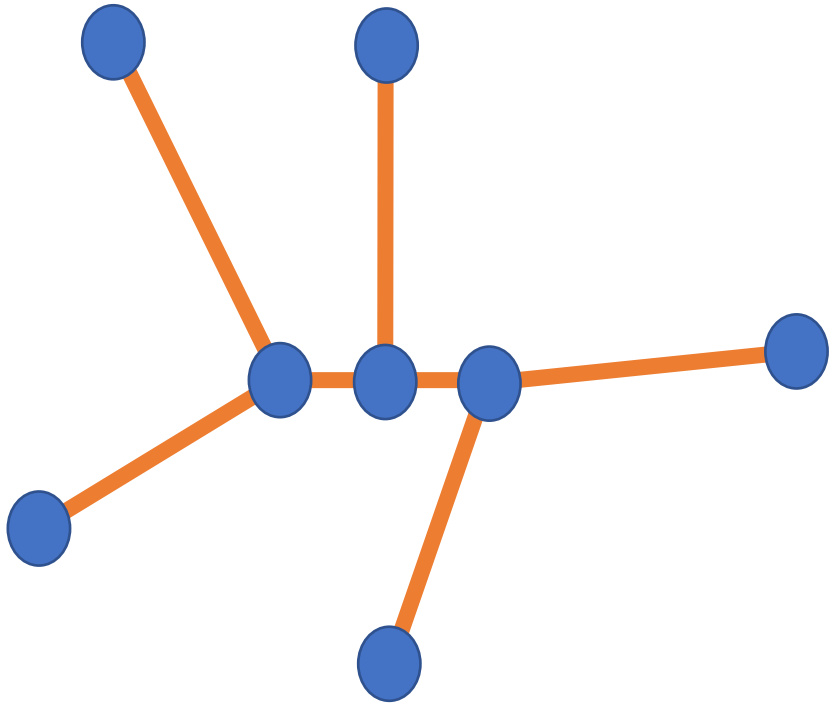
- Graph?
  - Nodes/edges?
  - What should we use node for?
  - What should we use edges for?
    - Node for junctions?
    - Edges for road segments connecting junctions?

Can we represent a road network using a mathematical model?

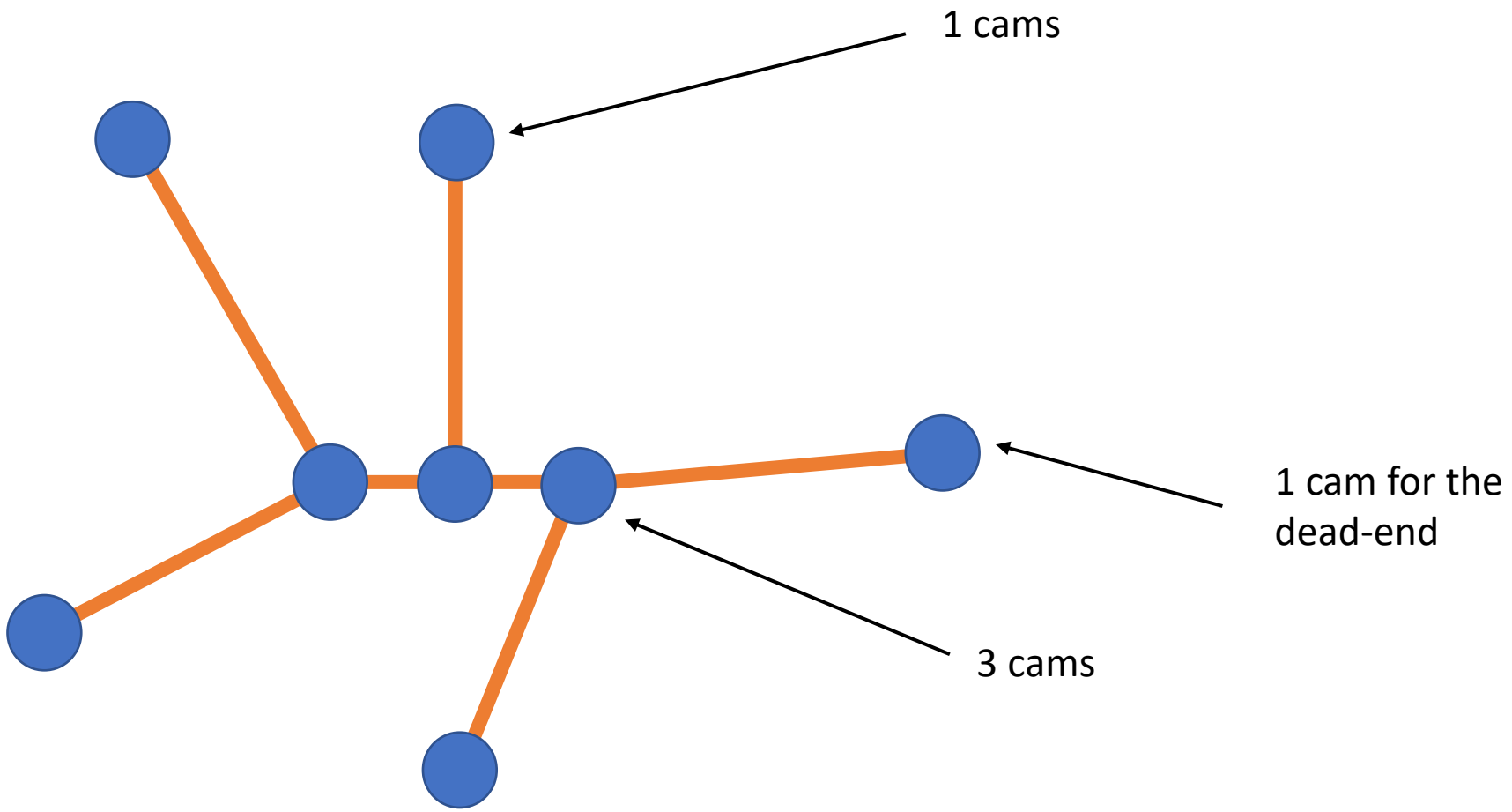
- Graph?
  - Nodes/edges?
  - What should we use node for?
  - What should we use edges for?
    - Node for junctions?
    - Edges for road segments connecting junctions?



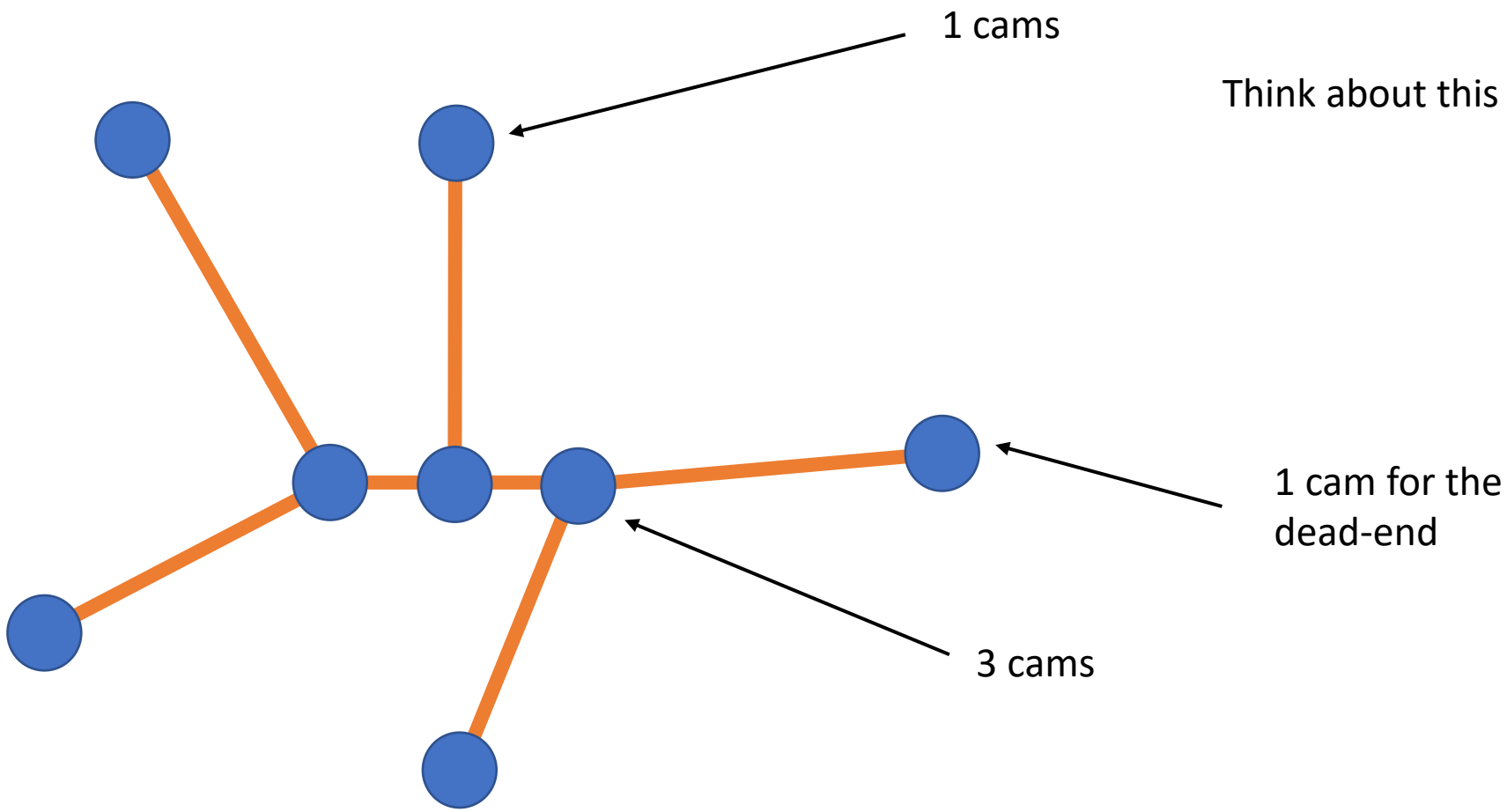
Now, let’s see whether we can answer questions using this model



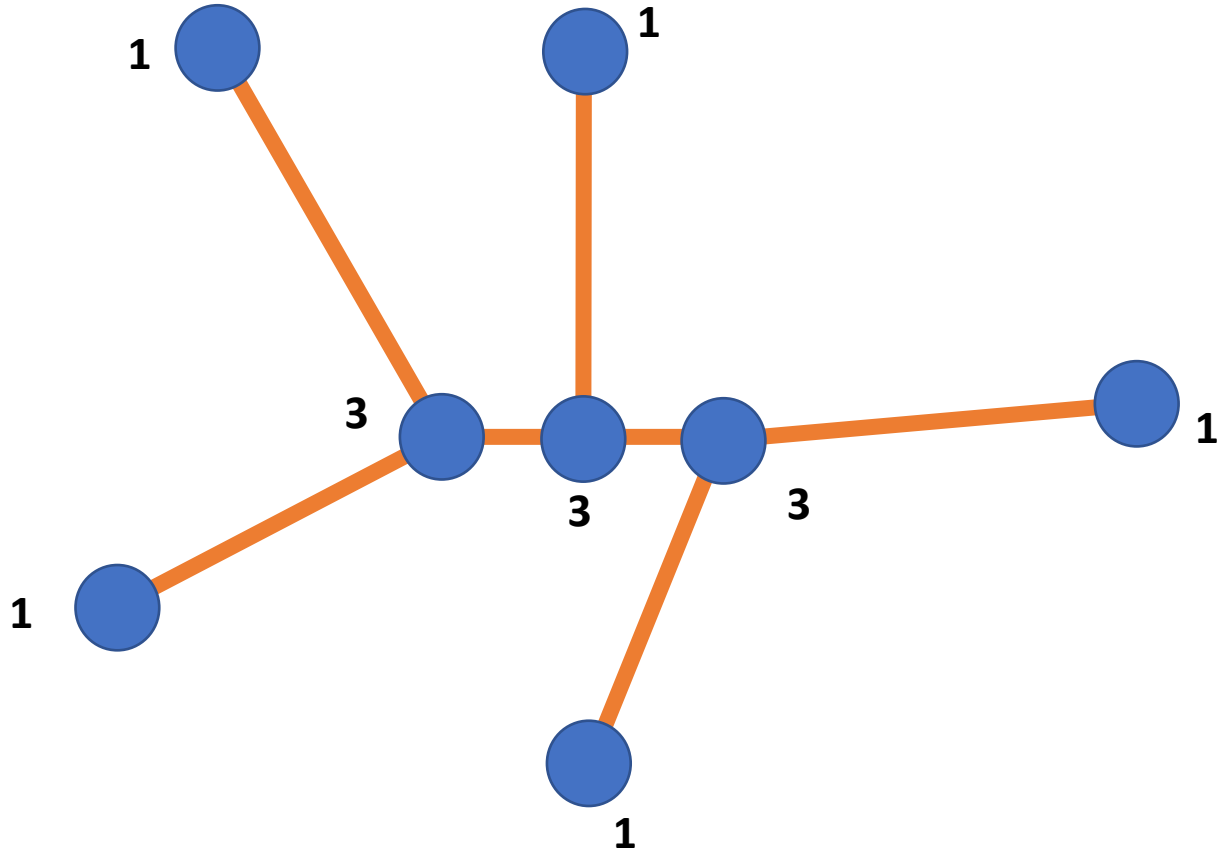
Each intersection requires a camera to monitor each road segment



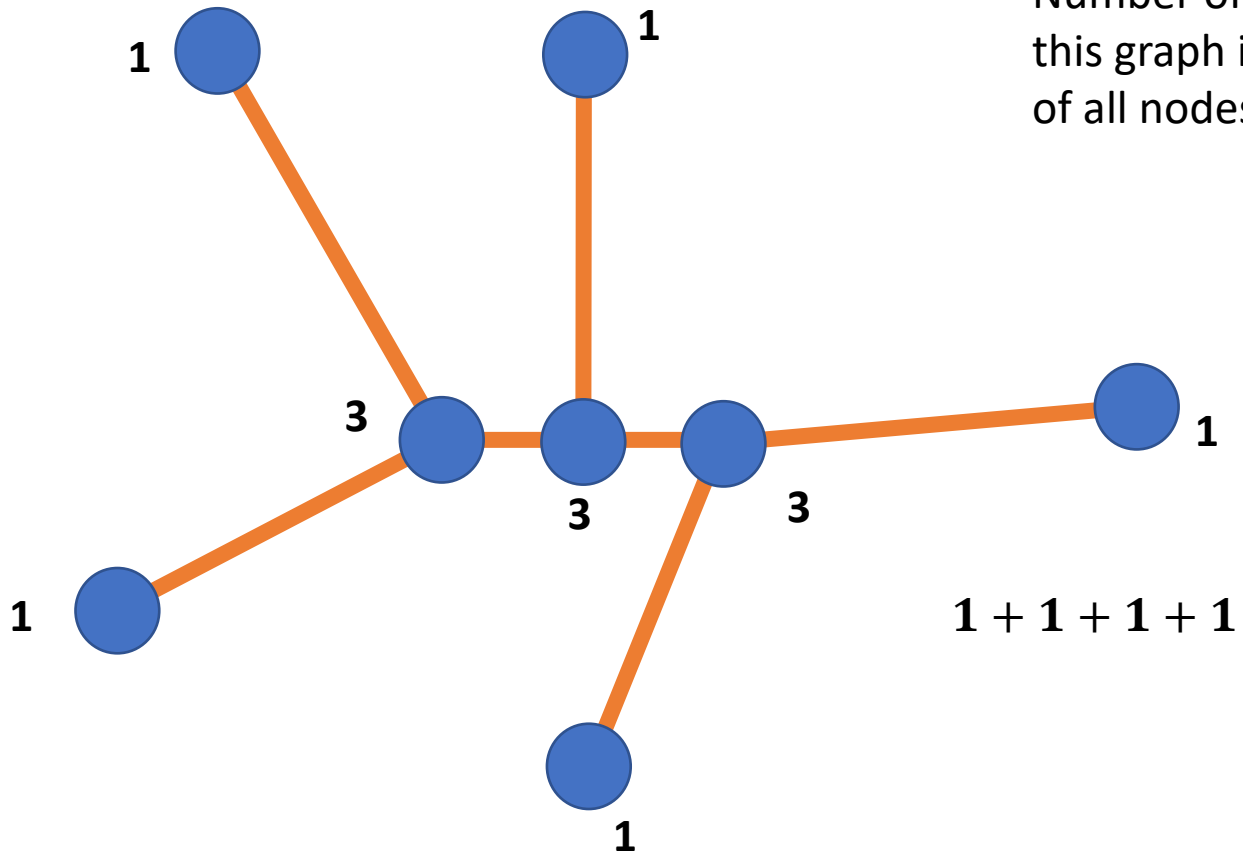
Can we build a road network such that required number of cameras is odd?



Can we build a road network such that required number of cameras is odd?



Can we build a road network such that required number of cameras is odd?



Number of cameras needed for this graph is the sum of degrees of all nodes in the graph.

$$1 + 1 + 1 + 1 + 3 + 3 + 3 + 1 = 14$$

Even

Can you answer this question for any road network?

- Suppose all road networks can be represented as a graph
- What can you say about the sum of the degrees of any graph?
- Each edge is considered twice when we consider the sum of degrees.
- Sum of degrees will always be even.
- Therefore, the answer to the question “Can we build a road network such that required number of cameras is odd?” is **NO**

Summary

- We considered an ill-defined thing.
- We built a formal model to represent it.
- Found limitations of the model, which translated to the limitations of the thing.
- This is basically what we are going to do in this class.

- We will learn fundamental capabilities and limitations of computers.
- We will essentially learn about 3 things.
  - Automata theory
    - Automata theory deals with the definitions and properties of mathematical models of computation.
    - Ex: finite automaton, context-free grammar
    - Formal definitions of computers and what each types of computers (models) can do and cannot do.
  - Complexity theory
    - We will try to classify problems based on how hard the problem is to solve.
    - **What makes some problems computationally hard and others easy?**
  - Computability theory
    - Study of whether a particular problem is solvable or not using a computer.
    - We will try to classify problems by those that are solvable and those that are not.

Textbook

- Introduction to the theory of computation by Michael Sipser.
- This is a theory course; you MUST read the book.

Communication

- Post your questions on MSU CSCI Discord channel.
  - CSCI-338-Computer-Science-Theory (CATEGORY)
- I will post the material on canvas and class GitHub page.
- I will use GitHub page for updating the schedule.
  - https://github.com/adiesha/CSCI338Fall2025

Pre-requisites

- CSCI 232 and CSCI 246
- If you have forgotten what you learned in Discrete Structures class, then go through the notes of Discrete Structures.
  - https://github.com/adiesha/CSCI246Fall2024
  - This is a link to my notes in the previous semester's discrete structure class.
- It is important that you understand the basics in Discrete Structures before taking CSCI 338
- I will do a recap of the basics in the first few lectures. (This is a quick overview; I will not take lot of time to explain)

Evaluation and grading

Grading	Weight of each category
Pop up Quizzes	4%
Assignments	42%
In class tests	30%
Final exam	20%
Attendance	4%
Total	100%

Evaluation and grading

- The final is optional. If you are happy with the grades of the 3 in-class tests, then the final test grade will be the average (in percentage) of the 3 in-class tests multiplied by 20. If you are not happy with the average of your in-class test, then you could choose to take the optional final.
- **The optional final cannot be taken early.**
- Note that quizzes are not an attendance counting, however, they will be graded lightly.
- At the end of the semester, final grades will be determined based on your overall performance on assignments, quizzes, tests and the final exam.

Accommodations

- If you have a documented disability and need or might need accommodations, please reach out to the Office of Disability Services as soon as possible and then talk to me.

Collaboration policy

- For your homework assignments, you may discuss the problem with your peers.
- However, you are not allowed to copy proofs from your peers or use generative AI to complete your assignments.
- If you use resources from the internet to solve problems, you MUST cite them.

Latex

- Assignments MUST be completed using LaTeX and submitted as a PDF. There will be no exceptions to this rule—grading handwritten proofs is extremely difficult, so I must enforce this for the sake of the TA's sanity.
- You can create a free Overleaf account, which is a free platform for editing and compiling complex LaTeX documents online. This is very similar to Google Docs but with LaTeX.
- Basic Overleaf latex tutorials.
- If you do not know the LaTeX code for a specific mathematical symbol, you can use the following website: https://detexify.kirelabs.org/classify.html Draw the symbol that you want; it will provide you the necessary code for that symbol and the packages that you have to import.
- Using LaTeX for mathematical writing is infinitely easier than using Word or handwriting the solutions once you get the hang of it.