CSCI~~1~~232:

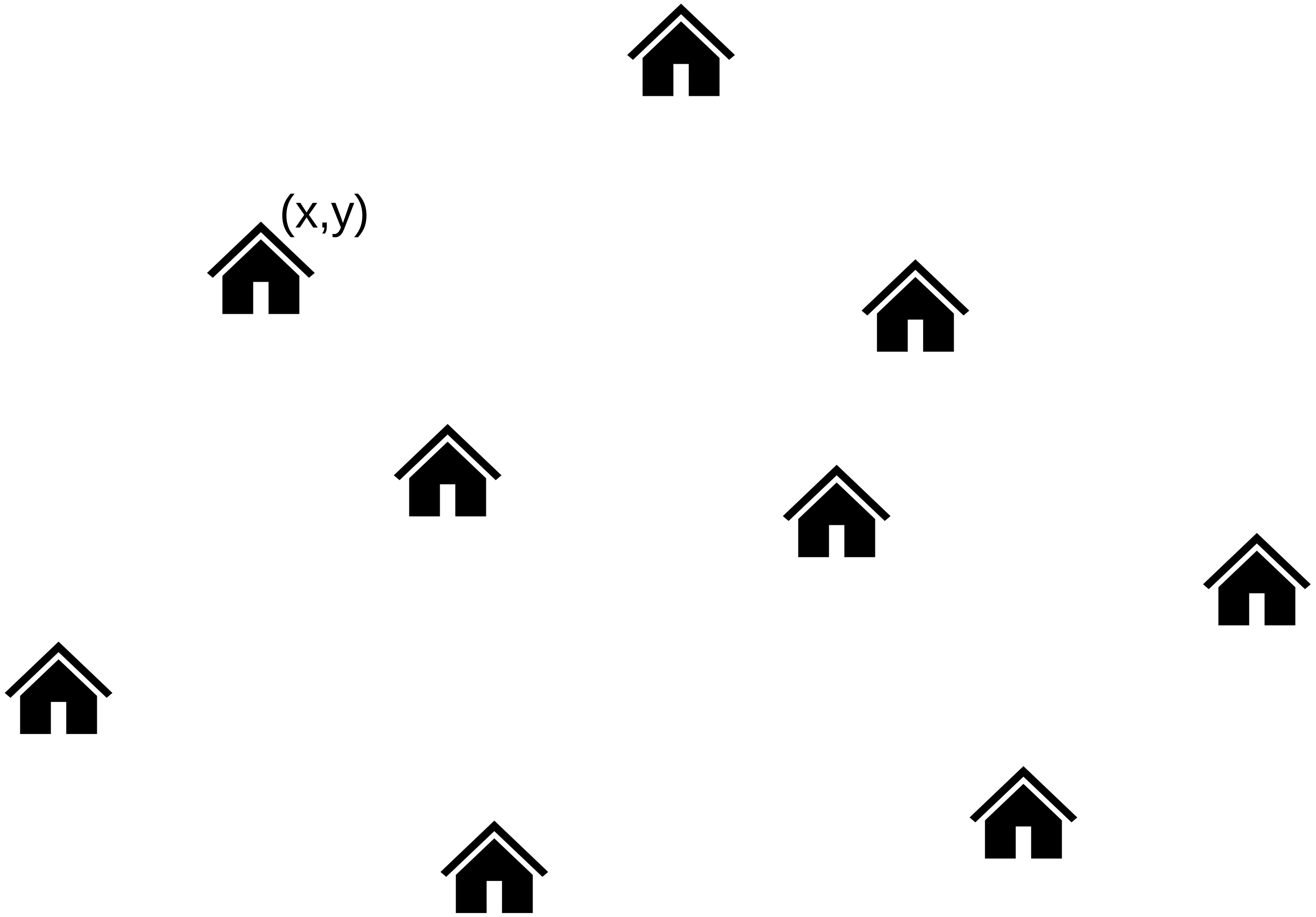
~~Basic~~ Data Structures and Algorithms

Course Intro, Syllabus, and Logistics

Reese Pearsall

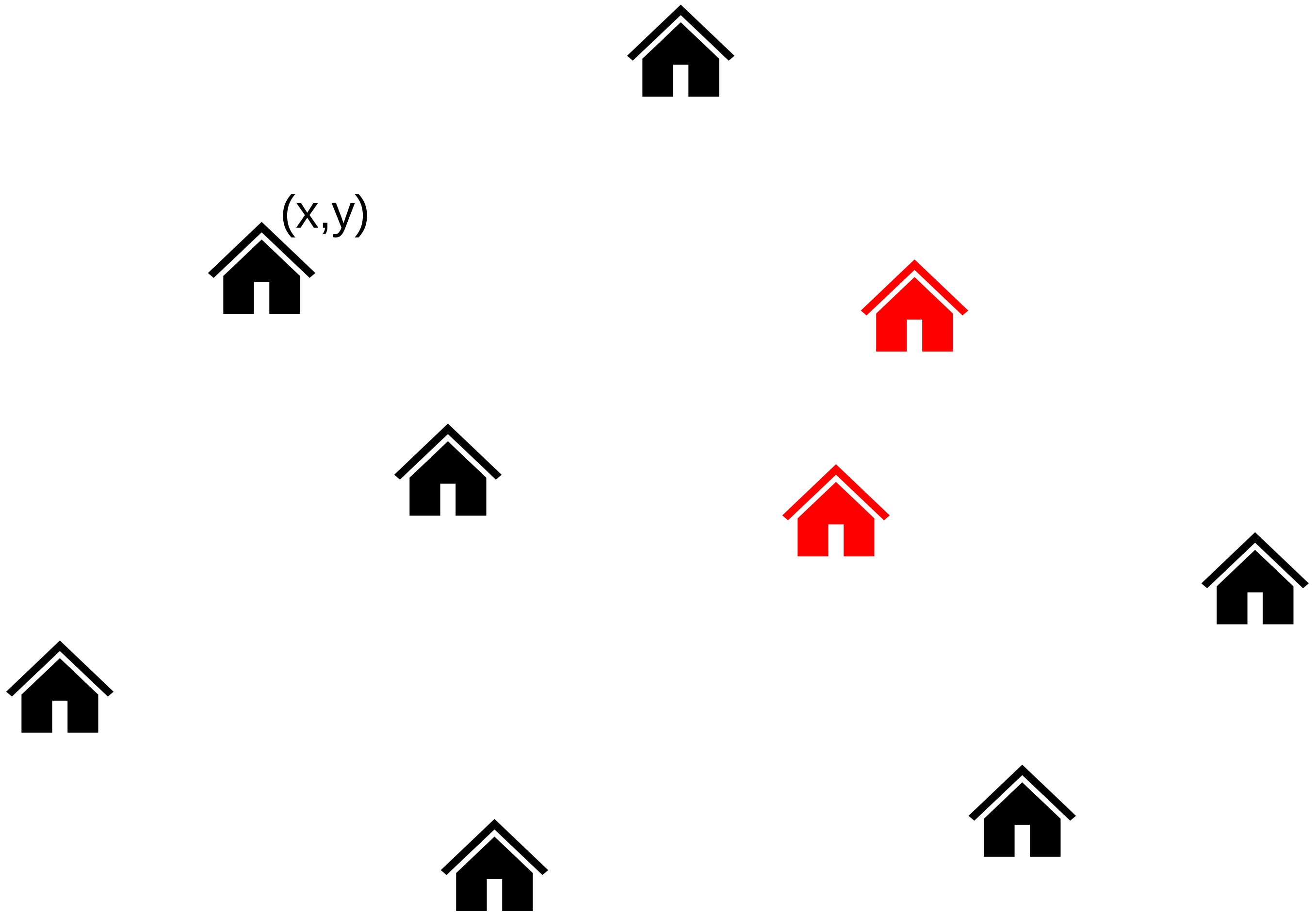
Spring 2024

To start things off, we are going to take a brief look at some of problems we are going to tackle in CSCI 232



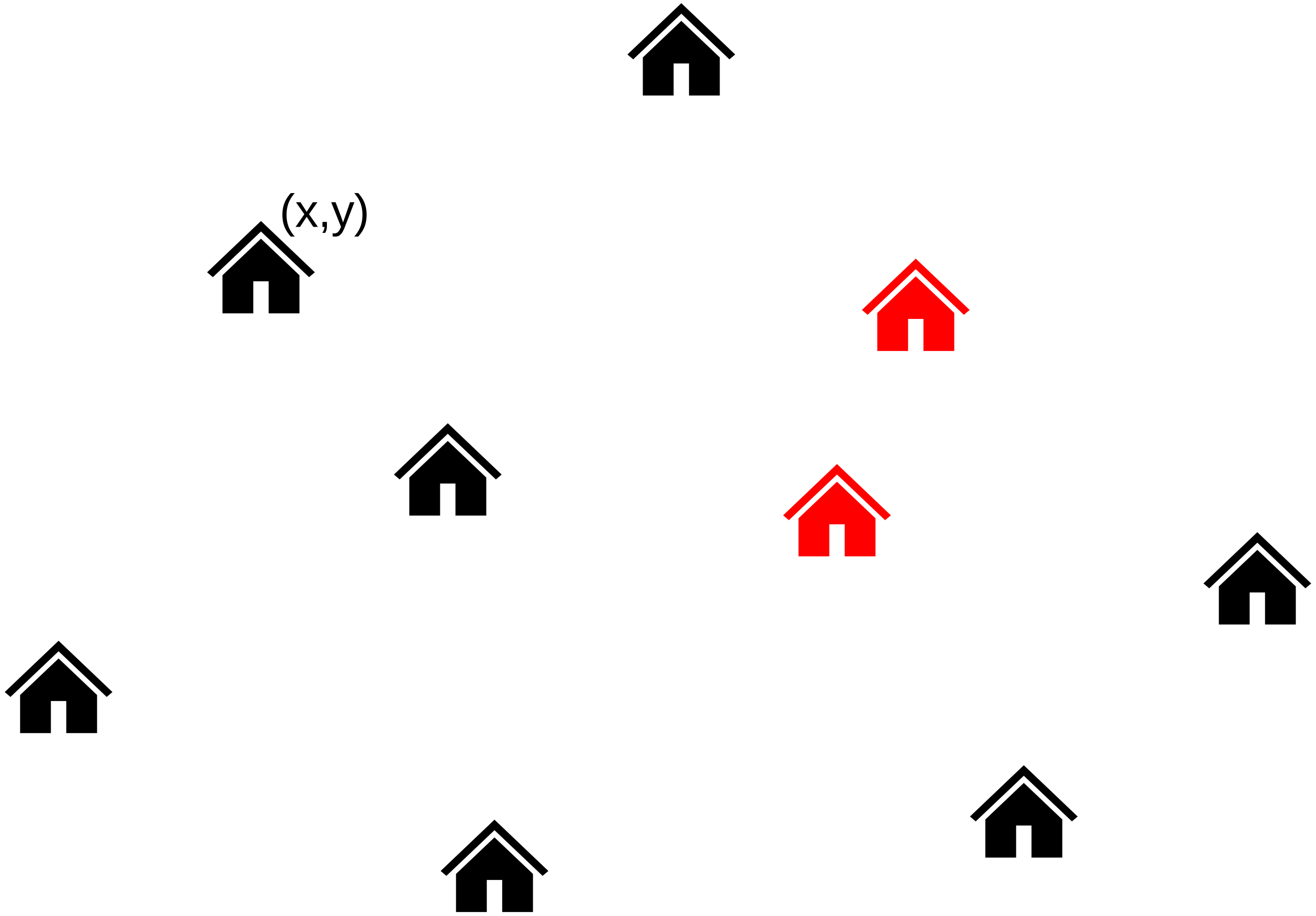
Given *n* houses, where each house has an x,y coordinate, find the pair of houses with the smallest distance between them

(You can assume no houses have the same x or y values)



Given *n* houses, where each house has an x,y coordinate, find the pair of houses with the smallest distance between them

(You can assume no houses have the same x or y values)

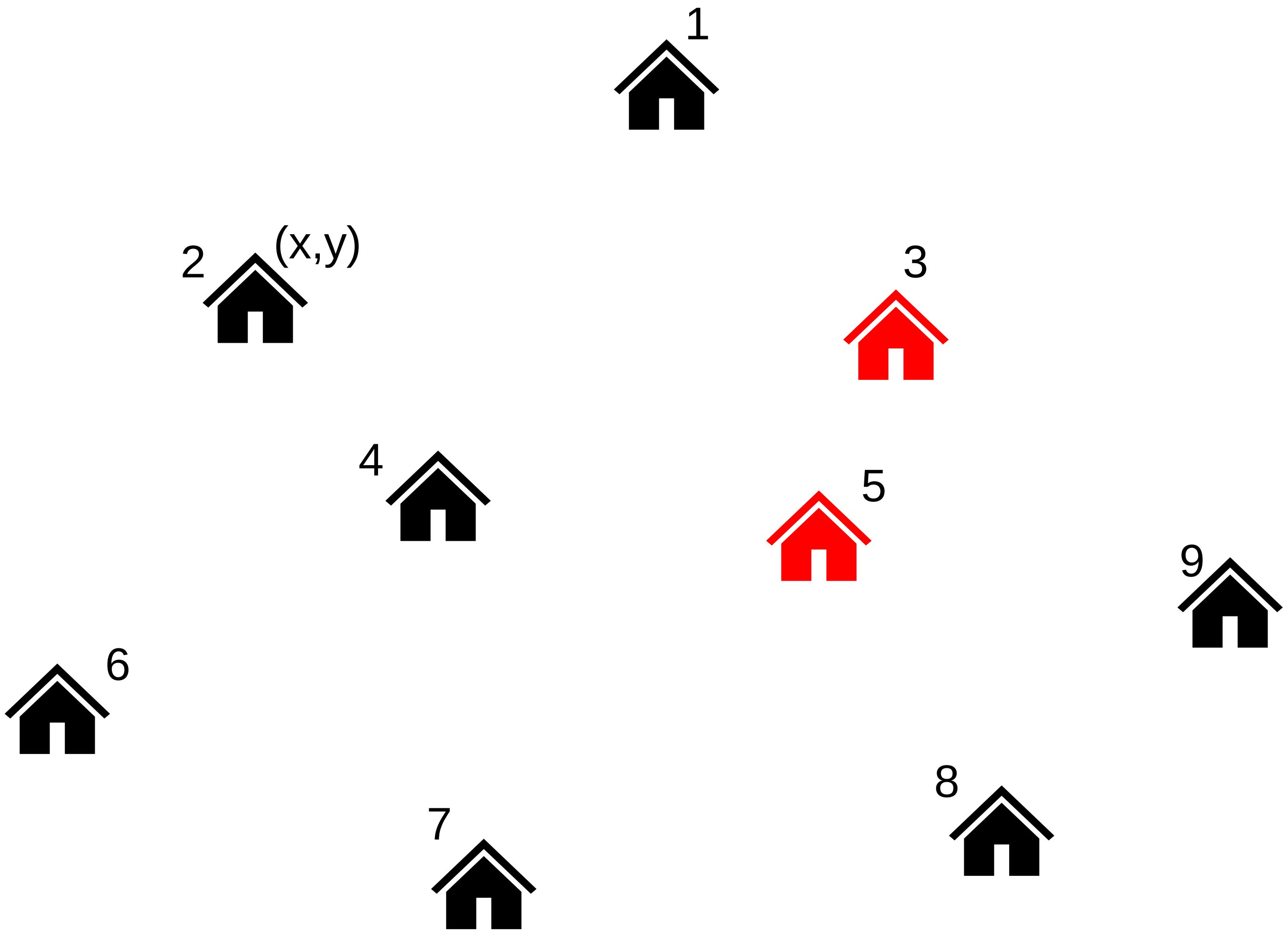


Given *n* houses, where each house has an x,y coordinate, find the pair of houses with the smallest distance between them

(You can assume no houses have the same x or y values)

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

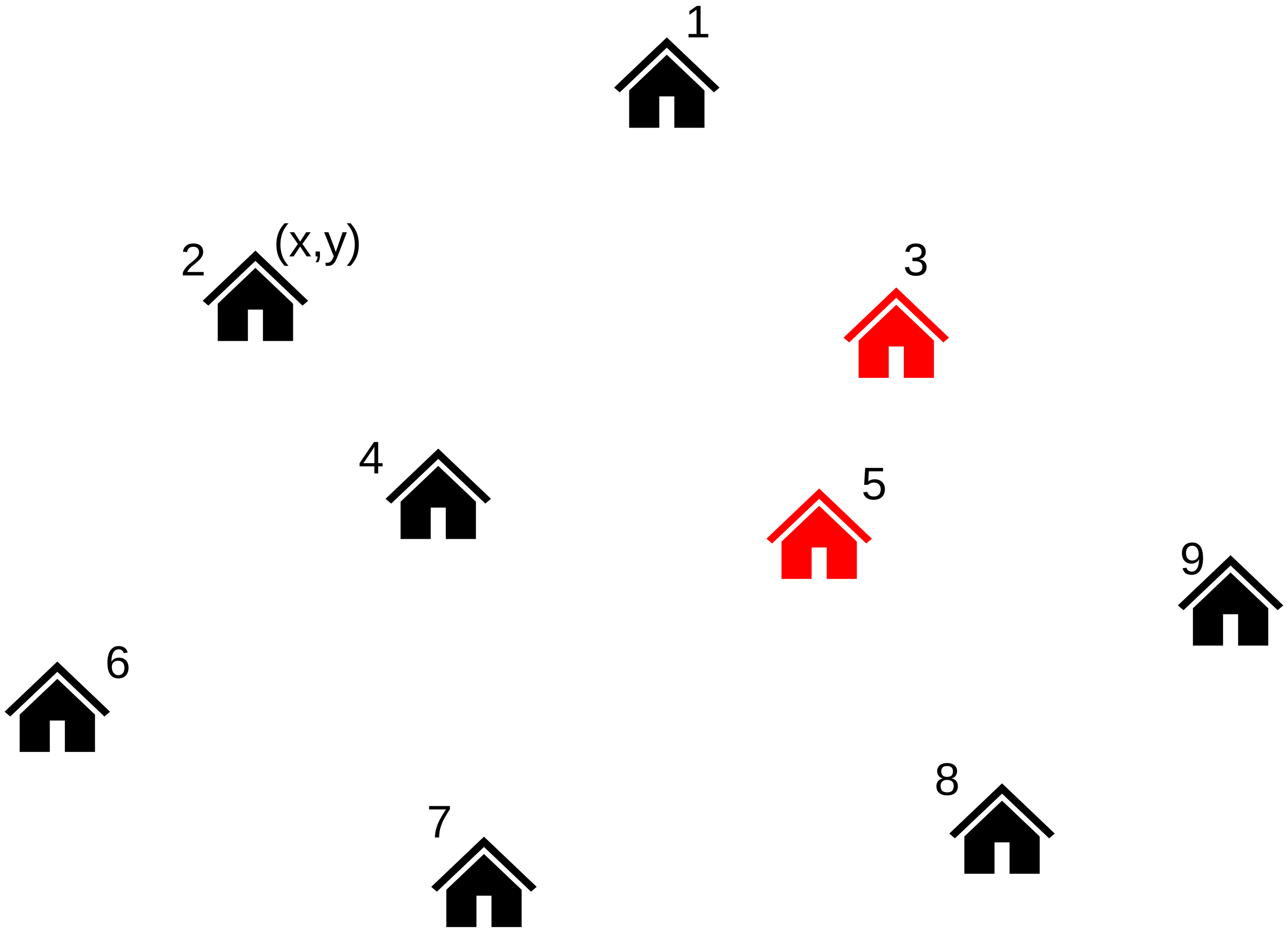
Algorithm ?



	H1	H2	H3	…	H9
H1	/	D(1,2)	D(1,3)	…	D(1,9)
H2	D(2,1)	/	D(2,3)	…	D(2,9)
H3	D(3,1)	D(3,2)	/	…	D(3,9)
…	…	…	…	…	….
H9	D(9,1)	D(9,2)	D(9,3)	…	/

Basic solution:

1. Compute distance for each pair
2. Select smallest

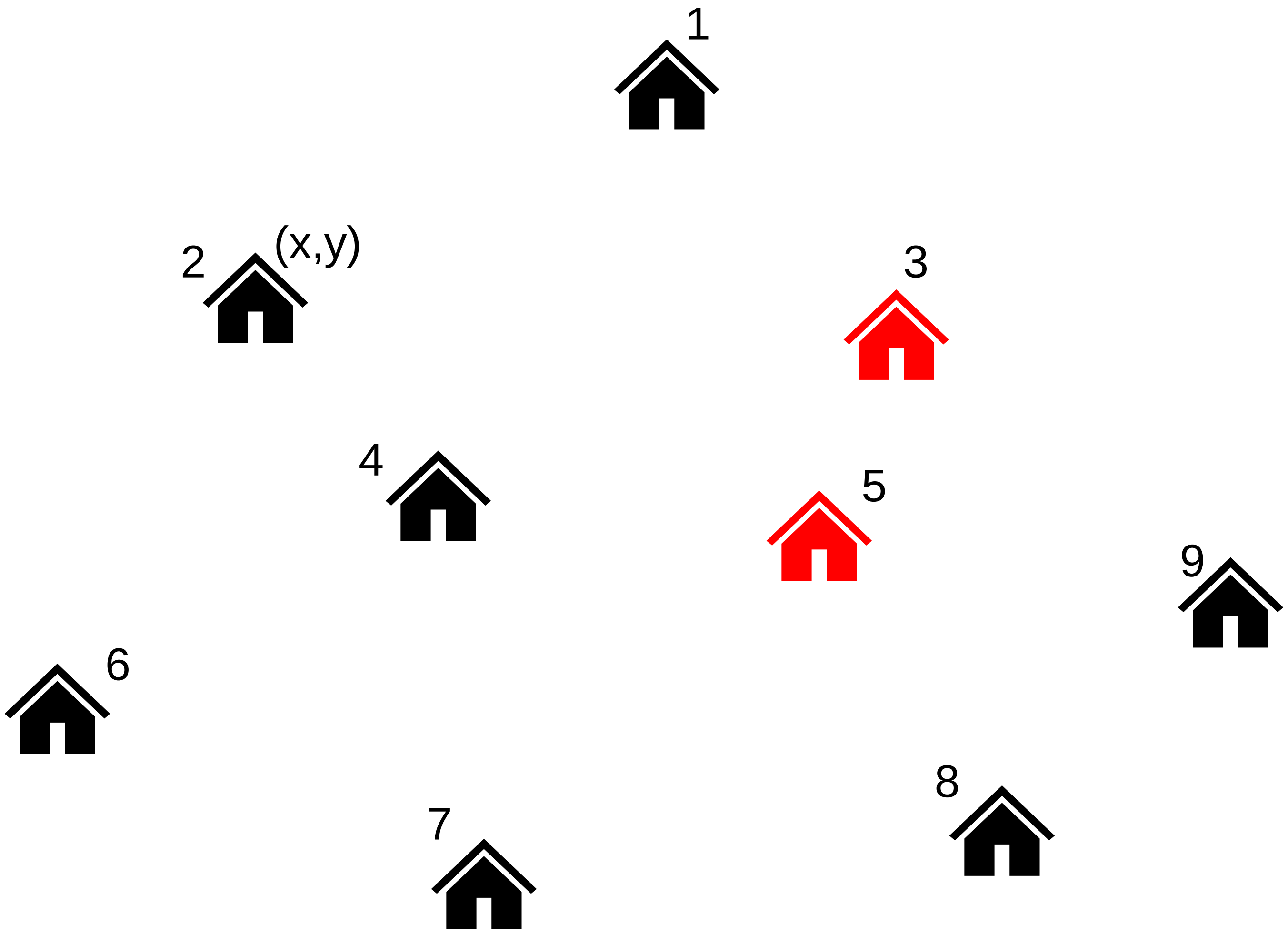


	H1	H2	H3	…	H9
H1	/	D(1,2)	D(1,3)	…	D(1,9)
H2	D(2,1)	/	D(2,3)	…	D(2,9)
H3	D(3,1)	D(3,2)	/	…	D(3,9)
…	…	…	…	…	….
H9	D(9,1)	D(9,2)	D(9,3)	…	/

Basic solution:

1. Compute distance for each pair
2. Select smallest

Running time = ?



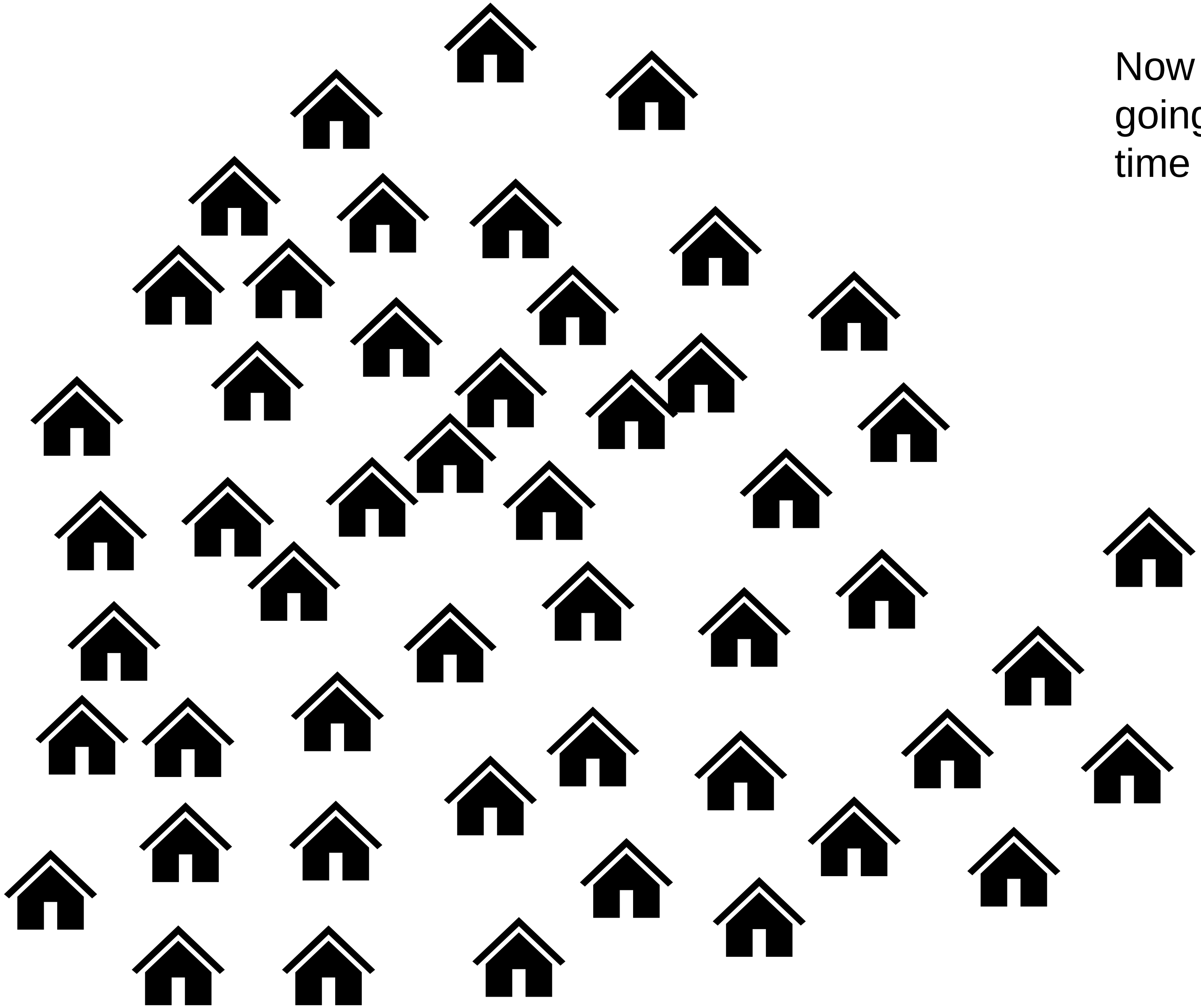
	H1	H2	H3	…	H9
H1	/	D(1,2)	D(1,3)	…	D(1,9)
H2	D(2,1)	/	D(2,3)	…	D(2,9)
H3	D(3,1)	D(3,2)	/	…	D(3,9)
…	…	…	…	…	….
H9	D(9,1)	D(9,2)	D(9,3)	…	/

Basic solution:

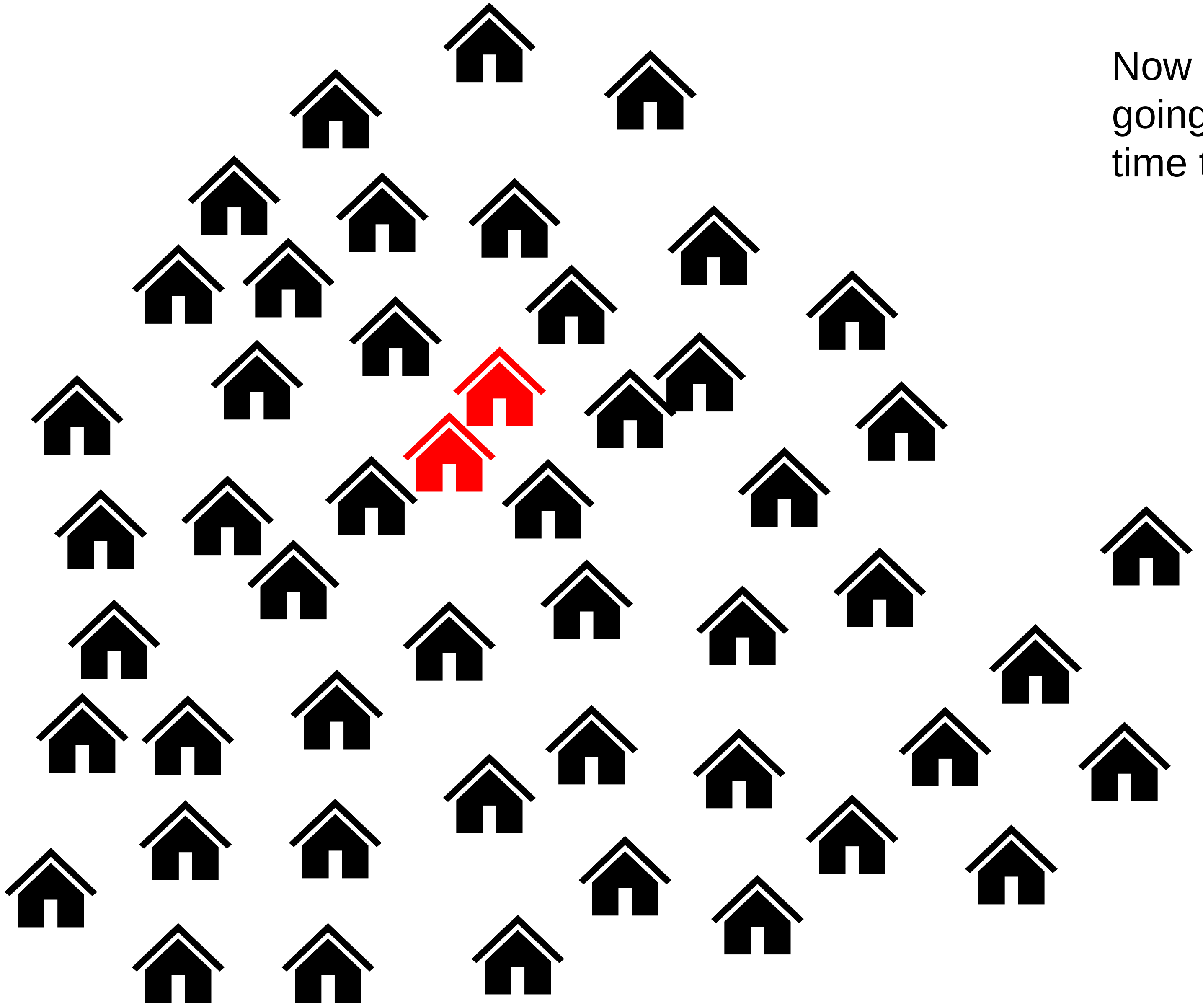
1. Compute distance for each pair
2. Select smallest

Running time = O(n^2)

(n = number of houses)

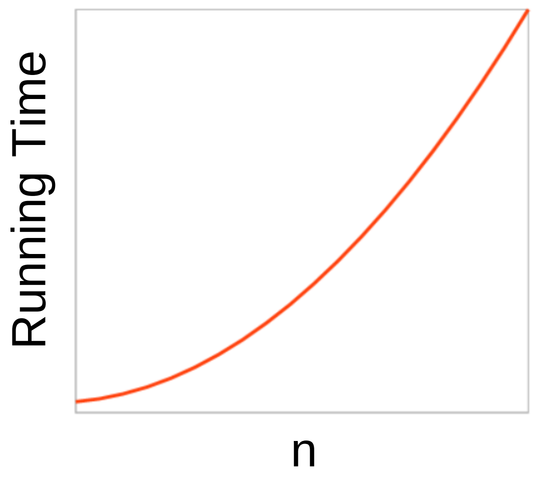


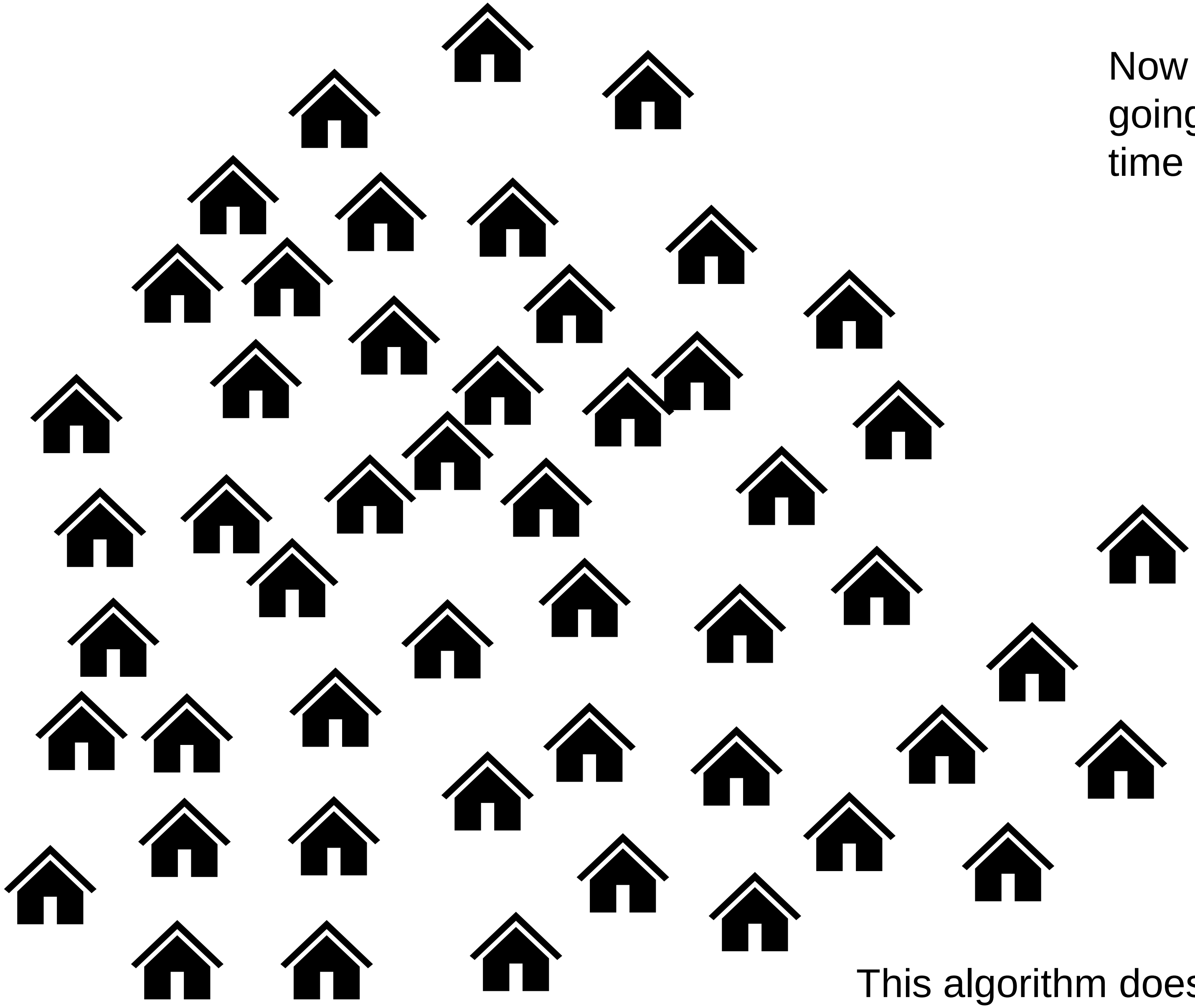
Now things get a bit messier. Our table is going to much larger, and will take more time to compute



Now things get a bit messier. Our table is going to much larger, and will take more time to compute

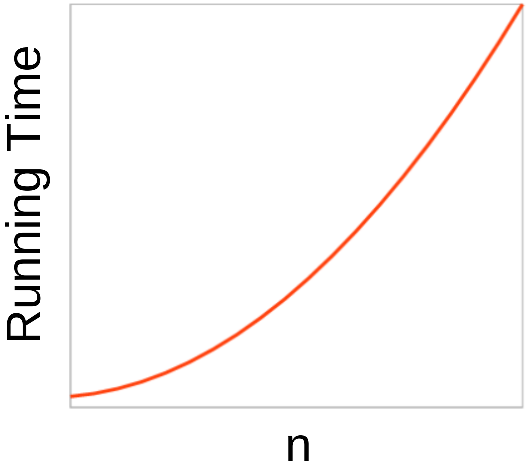
Remember, we are concerned with our algorithm performs **as some input *n* grows**





Now things get a bit messier. Our table is going to much larger, and will take more time to compute

Remember, we are concerned with our algorithm performs **as some input *n* grows**

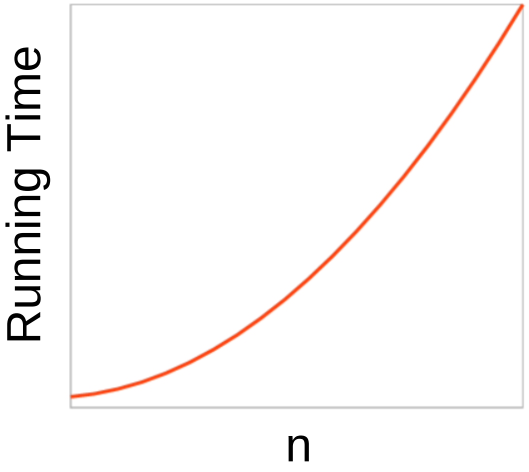


This algorithm does not perform very well as *n* increases

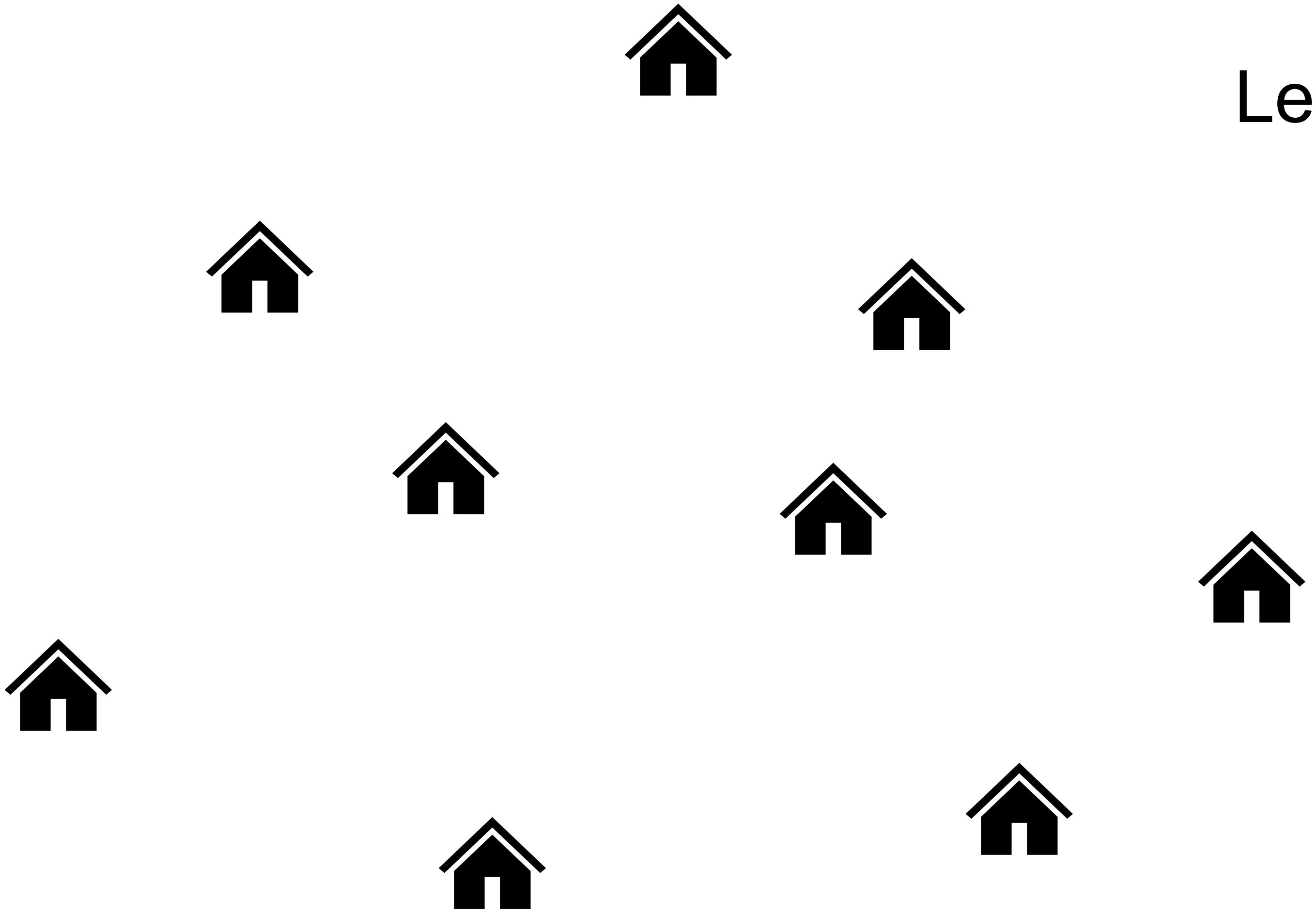
Now things g ur table is
going to ke more
ti

concerned with
erforms **as some**
grows

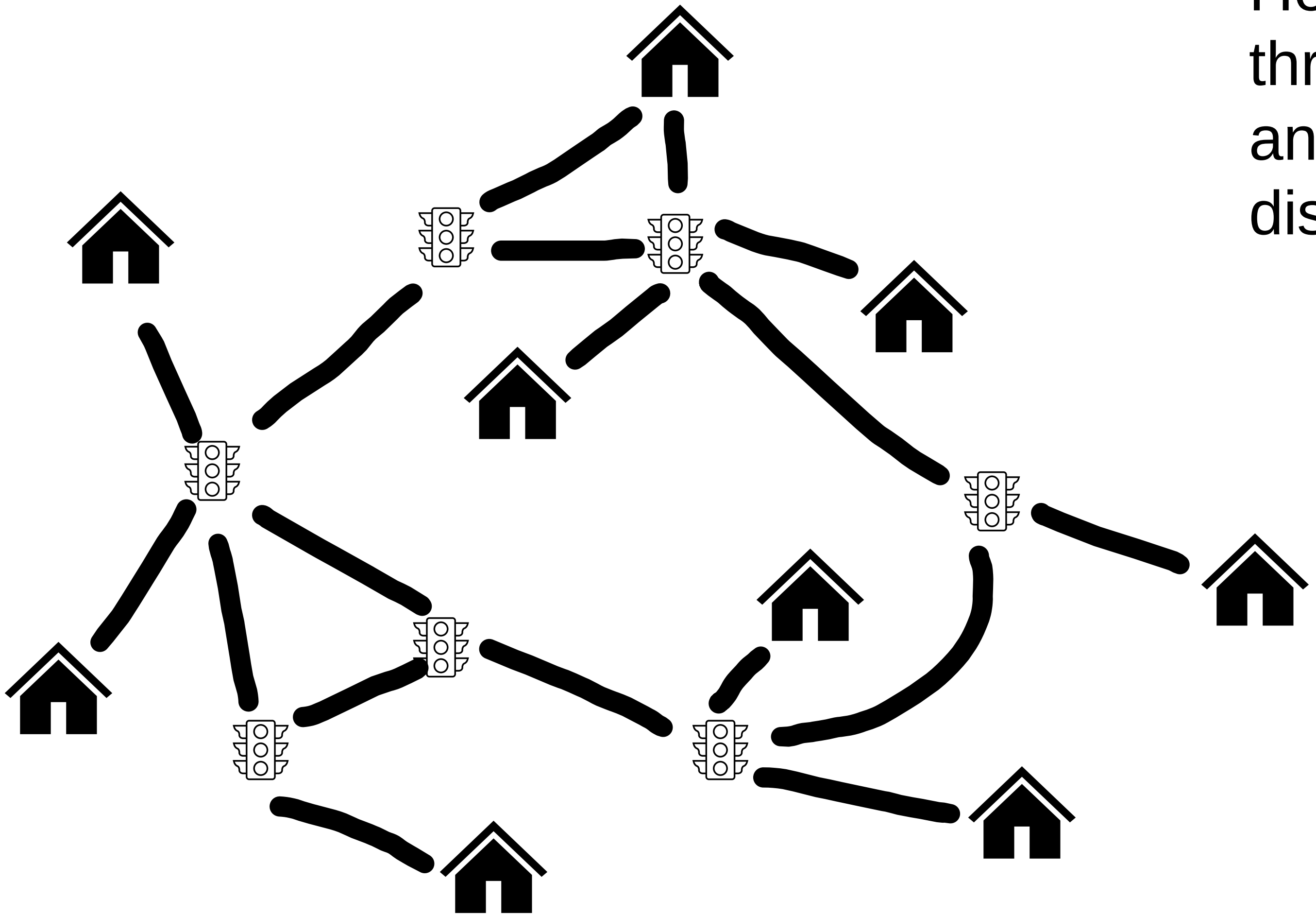
Can we do better?



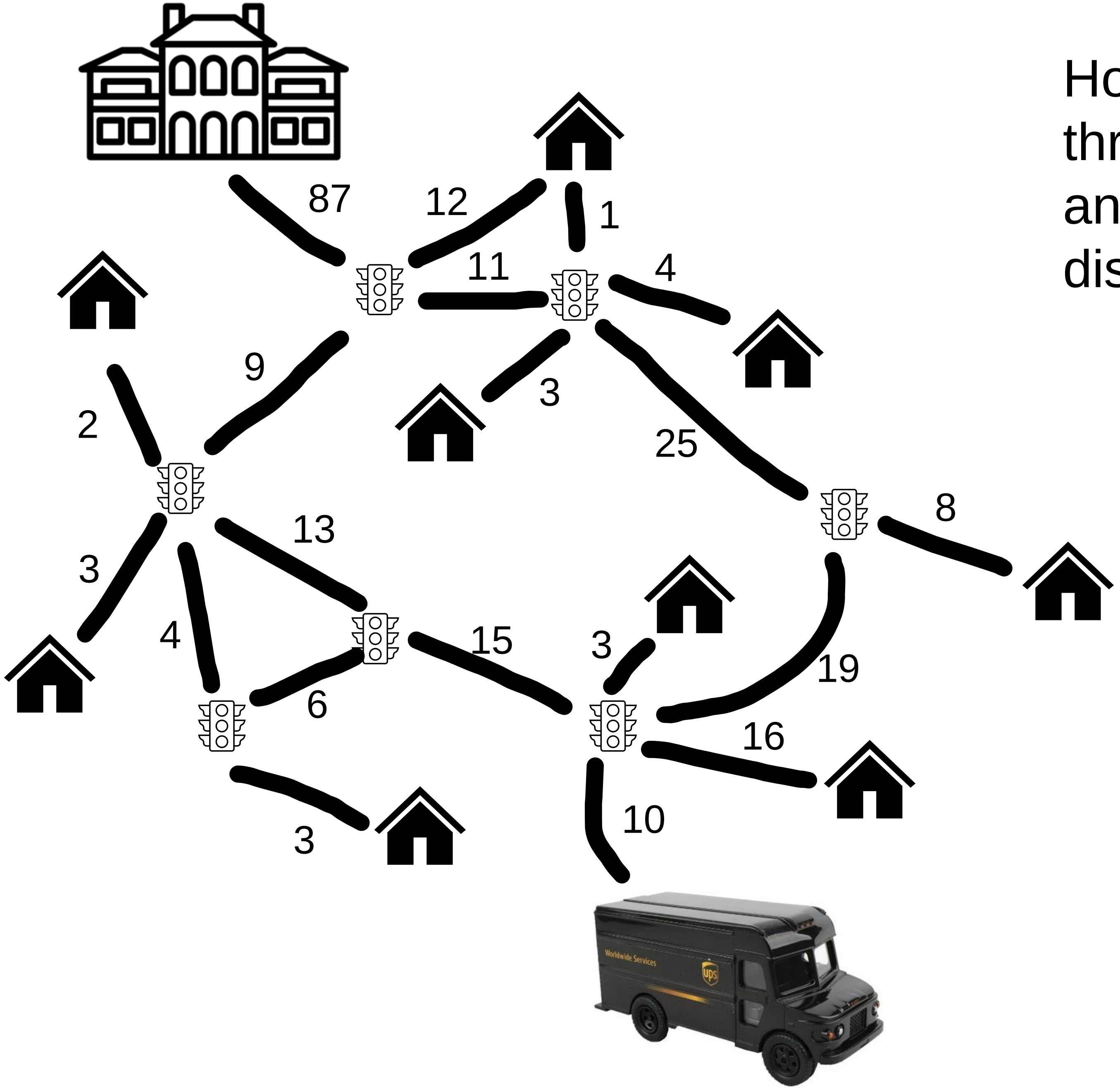
This algorithm does not perform very well as *n* increases



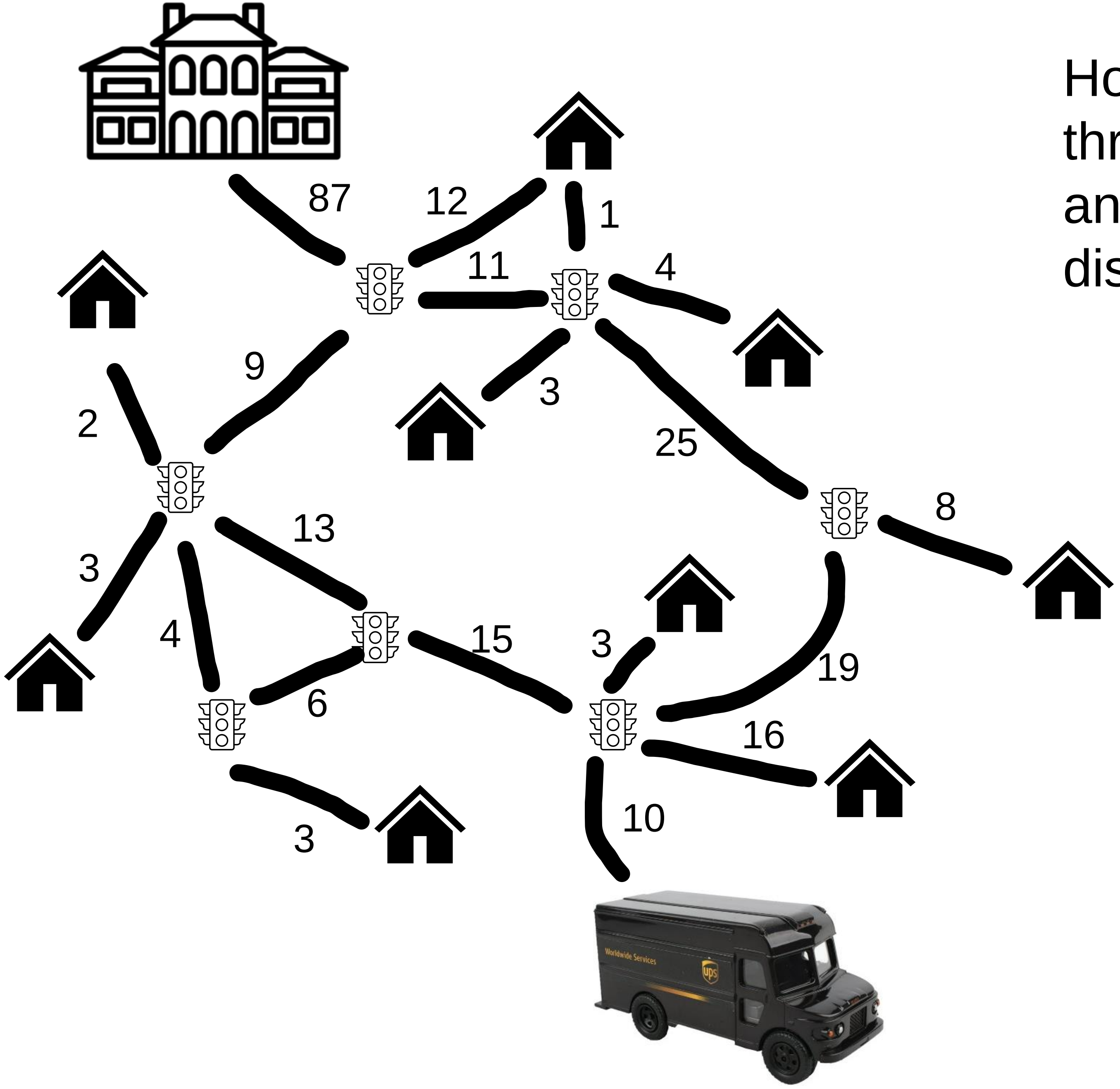
Let’s slightly tweak this problem



Houses are connected through a series of roads, and each road has some distance *d*

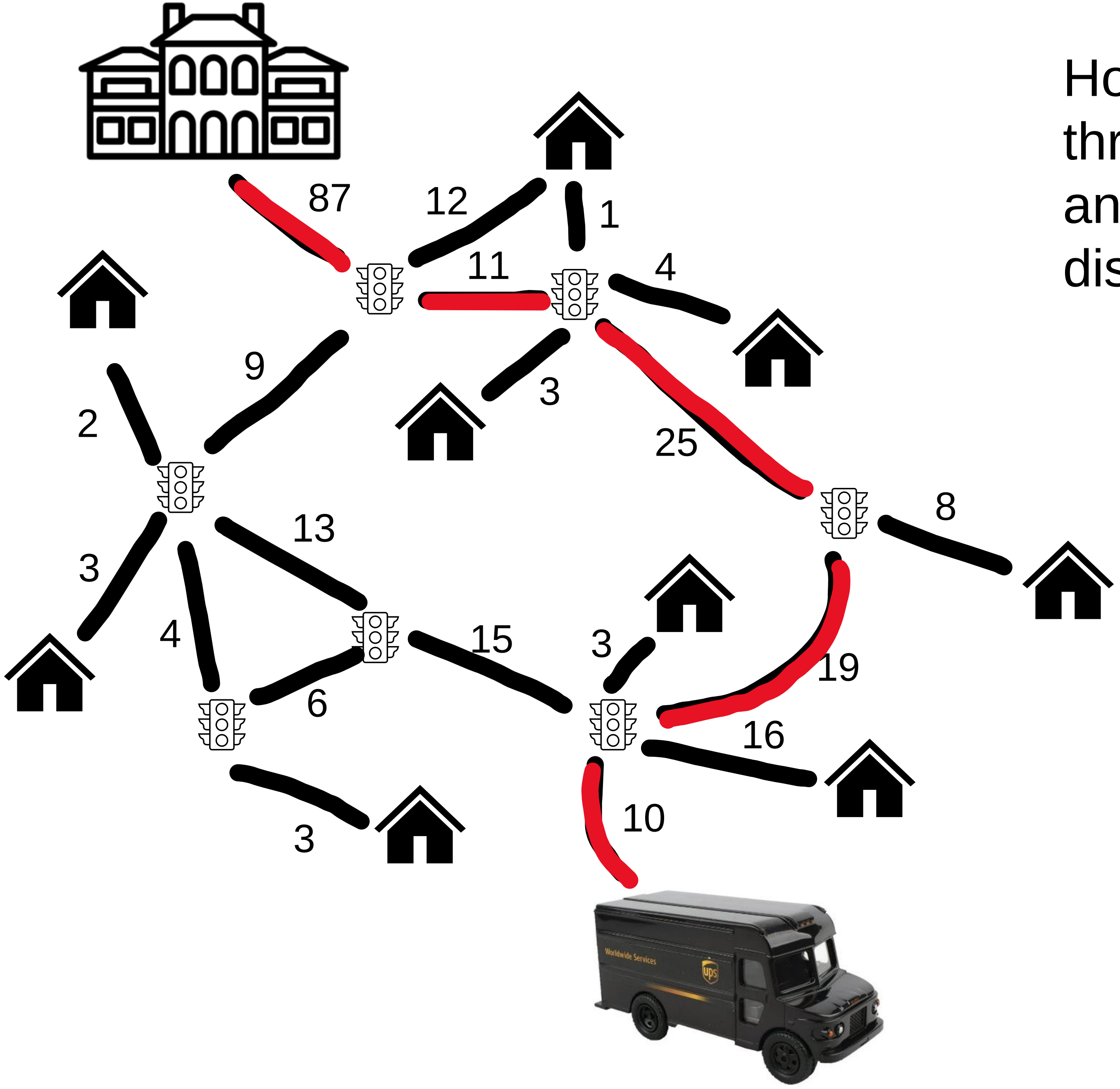


Houses are connected through a series of roads, and each road has some distance *d*



Houses are connected through a series of roads, and each road has some distance *d*

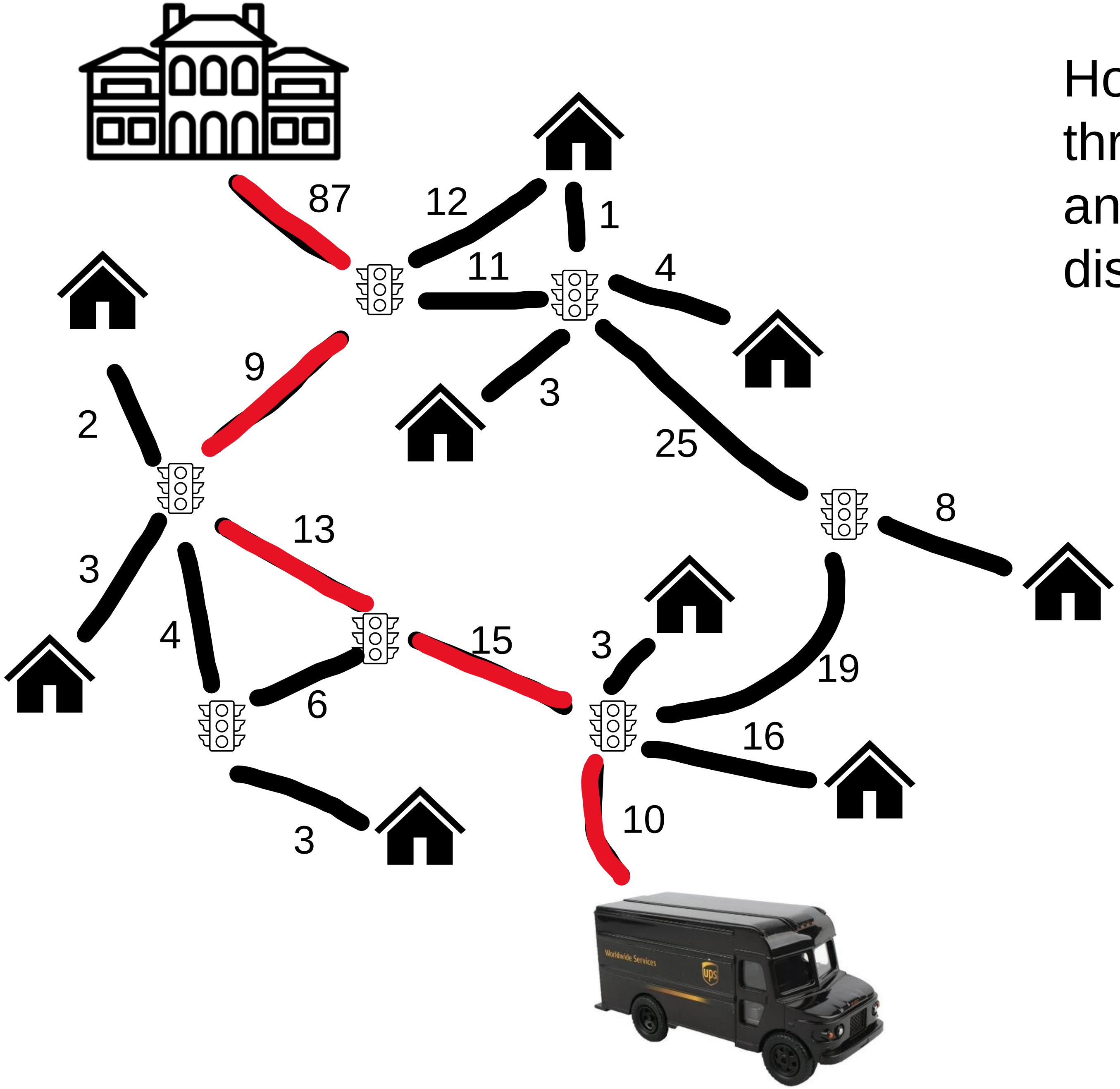
UPS person needs to deliver a package to the mansion



Houses are connected through a series of roads, and each road has some distance *d*

UPS person needs to deliver a package to the mansion

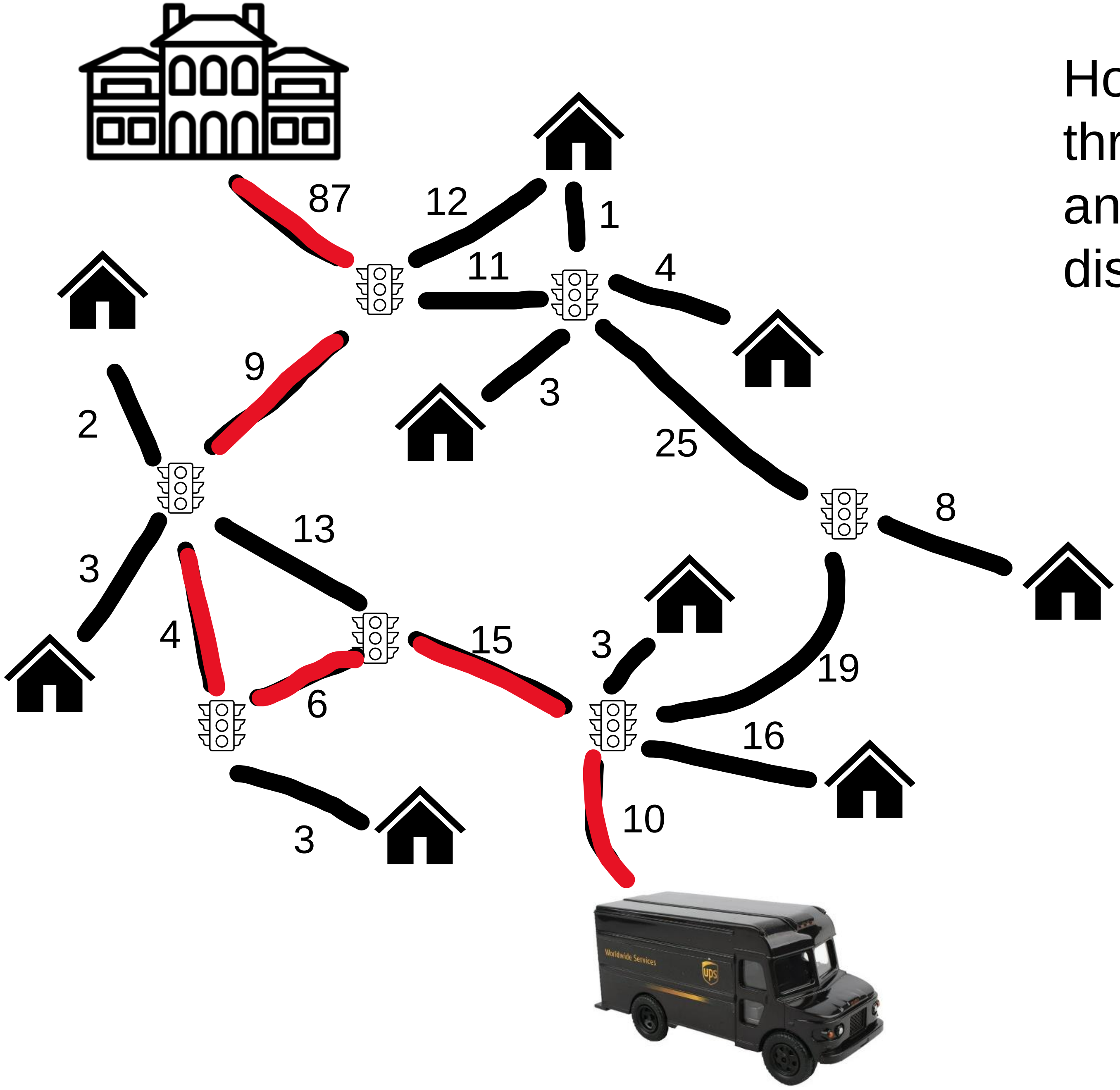
Cost: 152
(10 + 19 + 25 + 11 + 87)



Houses are connected through a series of roads, and each road has some distance *d*

UPS person needs to deliver a package to the mansion

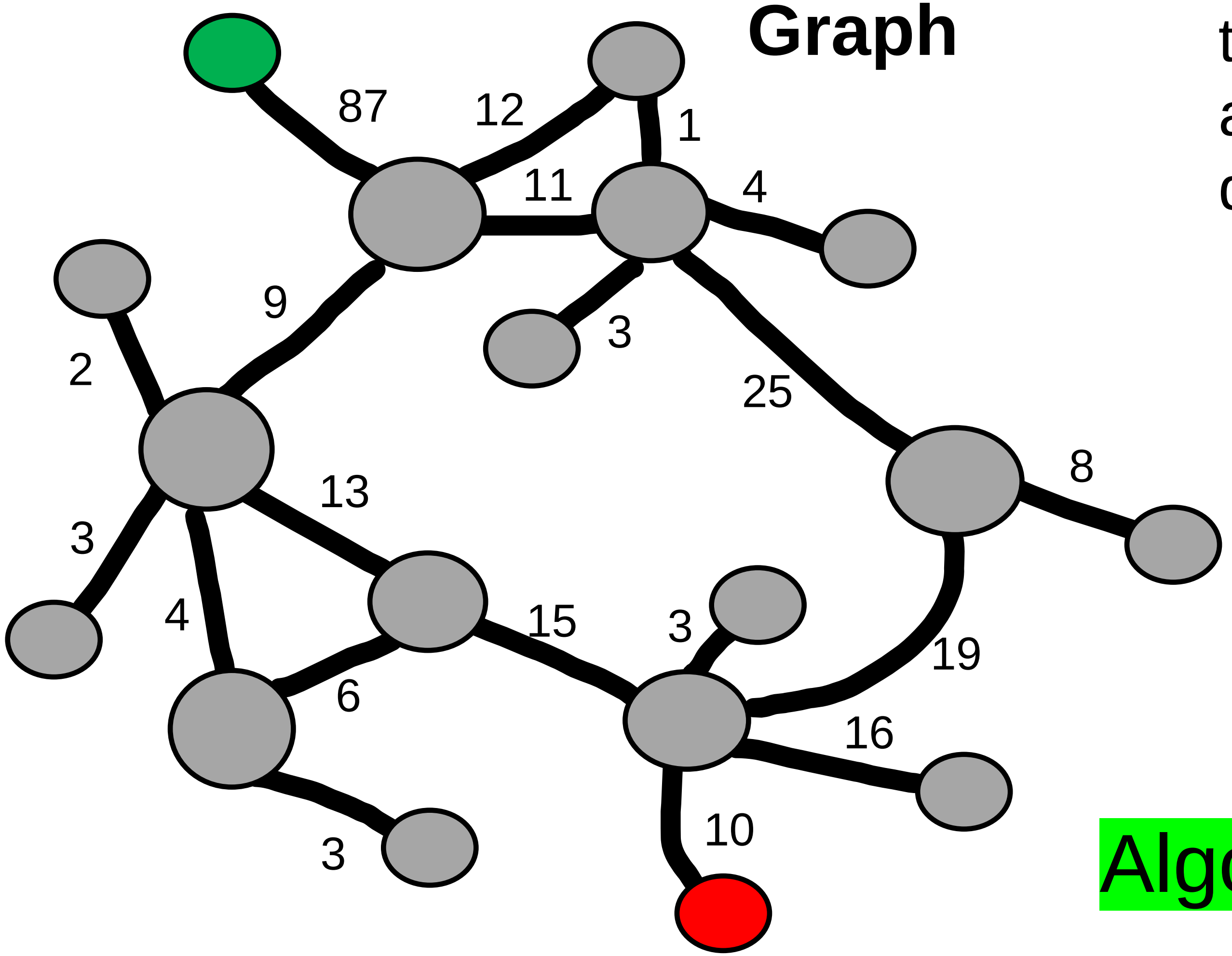
Cost: 134
(10 + 15 + 13 + 9 + 87)



Houses are connected through a series of roads, and each road has some distance *d*

UPS person needs to deliver a package to the mansion

Cost: 131
(10 + 15 + 4 + 6 + 9 + 87)



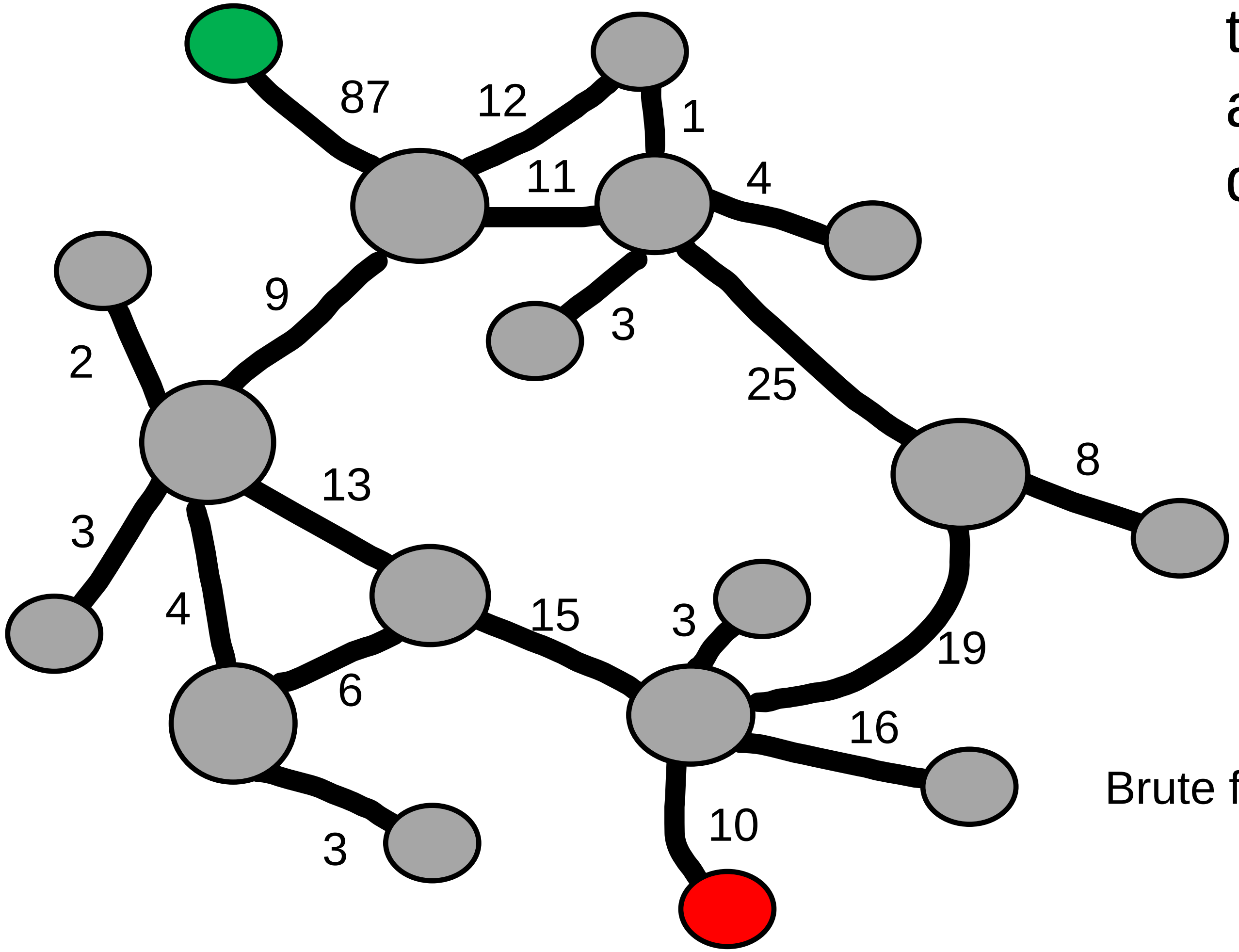
Graph

Houses are connected through a series of roads, and each road has some distance *d*

UPS person needs to deliver a package to the mansion

Goal: Find the **shortest path** from starting point (red) to ending point (green)

Algorithm ?

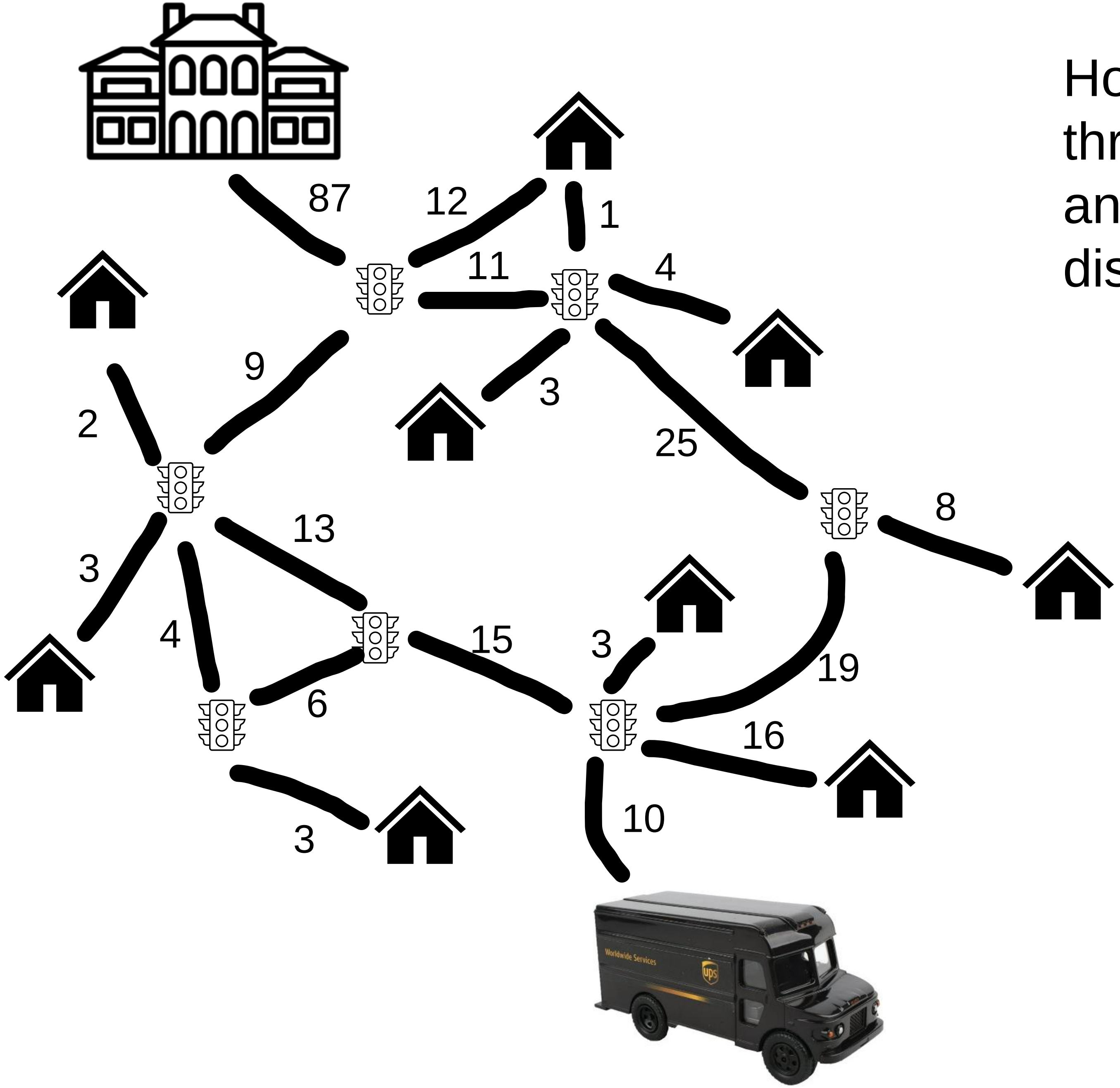


Houses are connected through a series of roads, and each road has some distance *d*

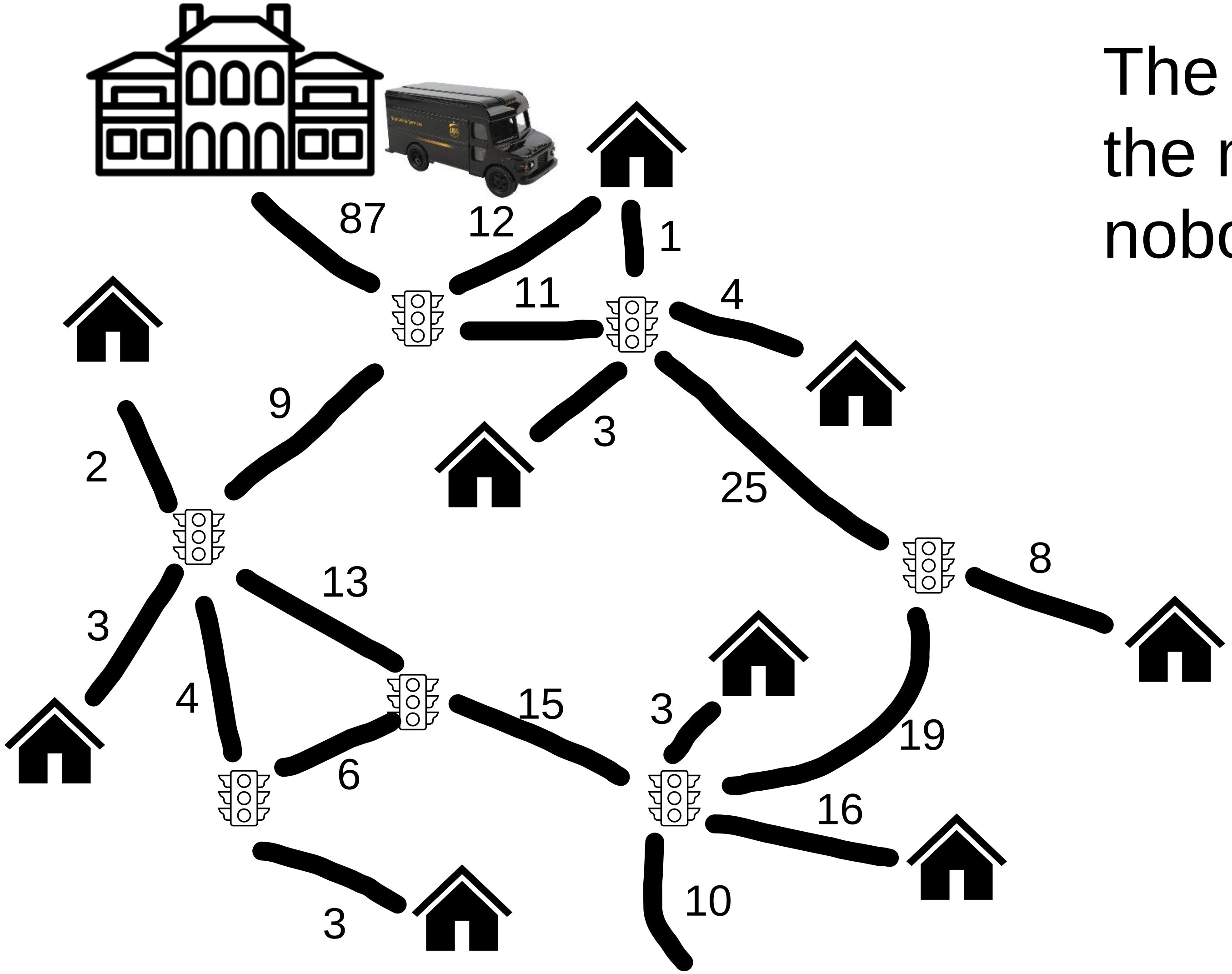
UPS person needs to deliver a package to the mansion

Goal: Find the **shortest path** from starting point (red) to ending point (green)

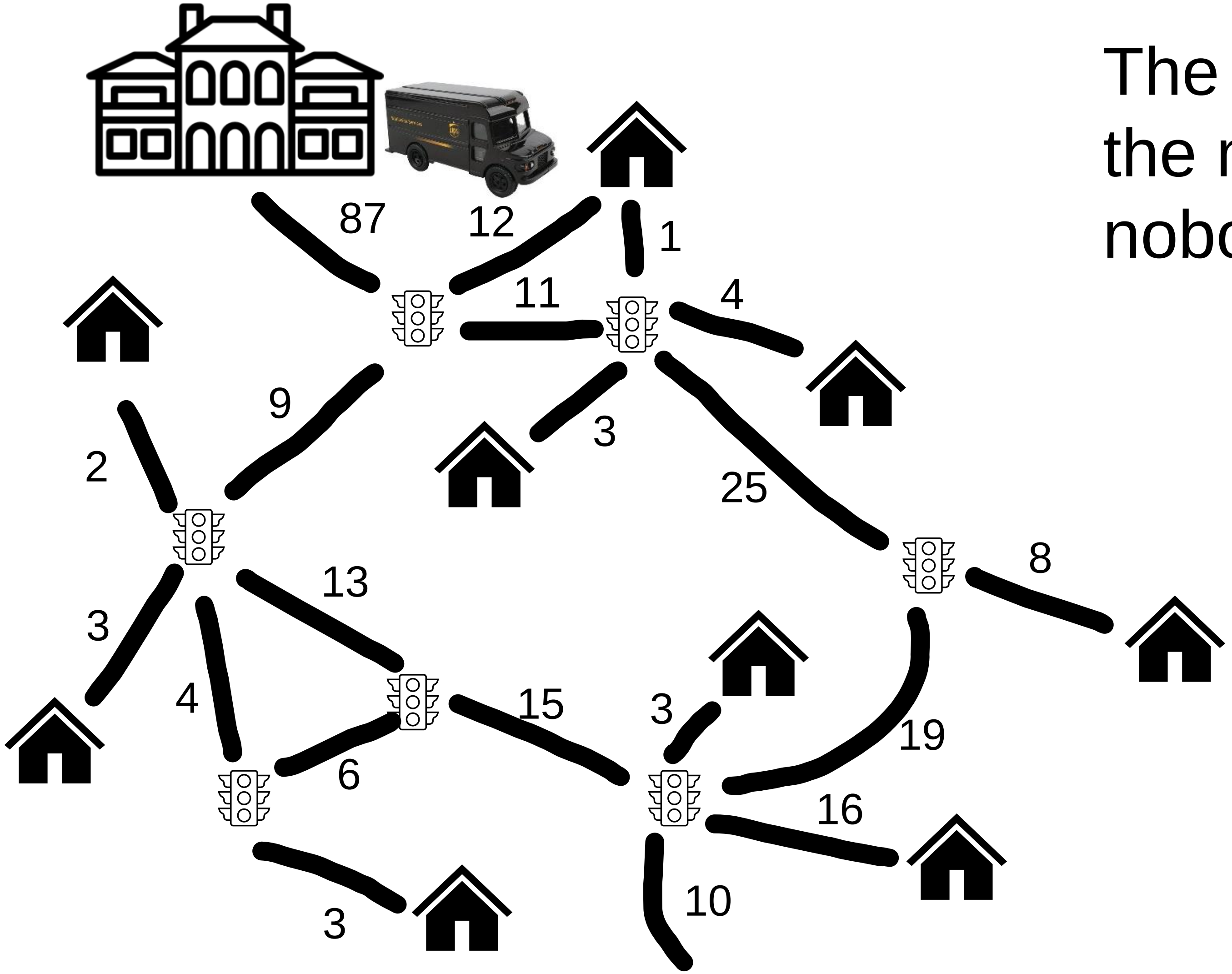
Brute forcing every possible path is **not feasible** (ie exponential or factorial time)



Houses are connected through a series of roads, and each road has some distance *d*



The delivery person arrives at the mansion, and notices that nobody is home



The delivery person arrives at the mansion, and notices that nobody is home

They decide they are going to **rob** the house and brings their backpack along

We are going to steal some items from the house and put them into our backpack

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items from the house and put them into our backpack

Each item has a weight, and a value

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

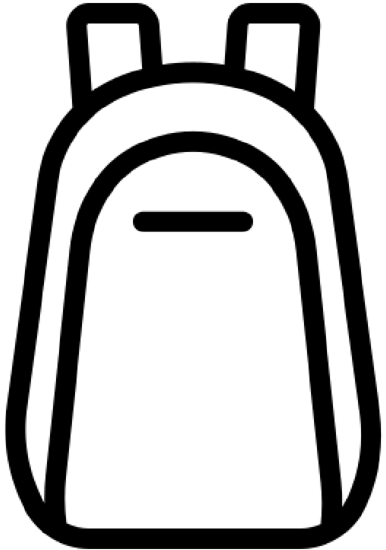
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items from the house and put them into our backpack

Each item has a weight, and a value

Goal: Steal items such that we **maximize the value** of items being stole, while **not overfilling our backpack**



Our backpack can only fill 10 pounds

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

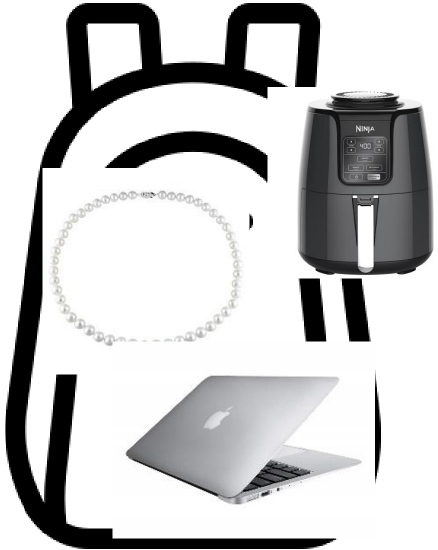
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items from the house and put them into our backpack

Each item has a weight, and a value

Goal: Steal items such that we **maximize the value** of items being stole, while **not overfilling our backpack**



Total Value: 70
Weight: 10

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

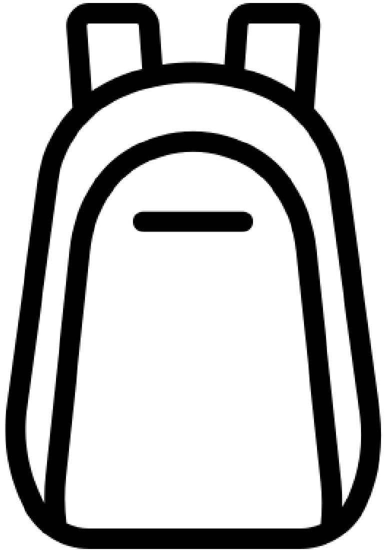
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items from the house and put them into our backpack

Each item has a weight, and a value

Goal: Steal items such that we **maximize the value** of items being stole, while **not overfilling our backpack**



Algorithm?

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

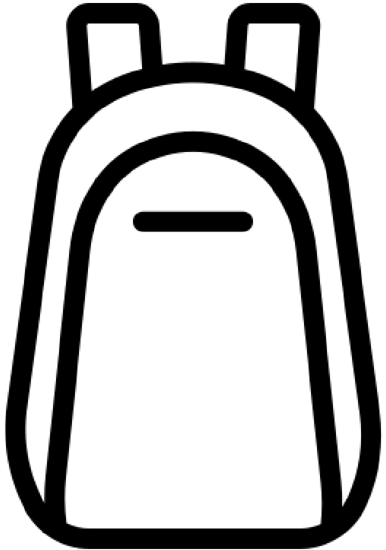
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items from the house and put them into our backpack

Each item has a weight, and a value

Goal: Steal items such that we **maximize the value** of items being stole, while **not overfilling our backpack**



Stuff our backpack with the most expensive items until we can’t fit anymore

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

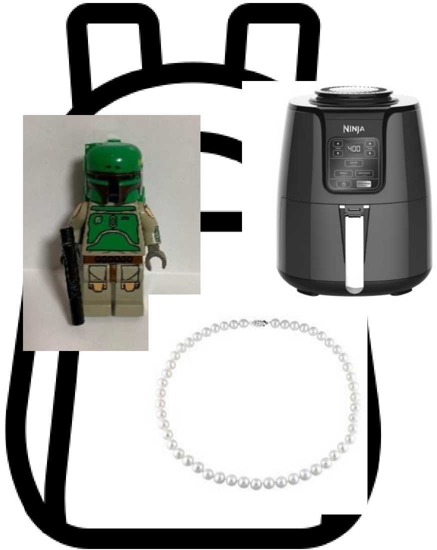
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items from the house and put them into our backpack

Each item has a weight, and a value

Goal: Steal items such that we **maximize the value** of items being stole, while **not overfilling our backpack**



Value: 90
Weight: 9

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

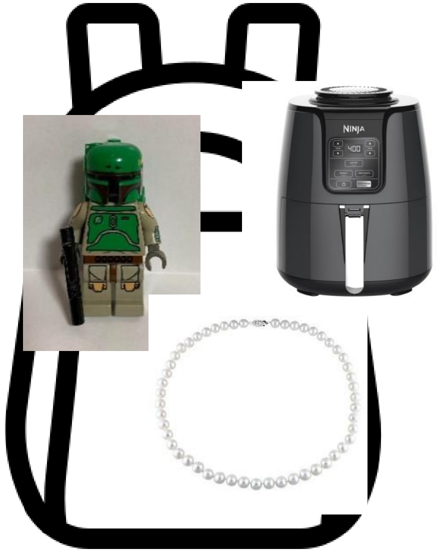
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items from the house and put them into our backpack

Each item has a weight, and a value

Goal: Steal items such that we **maximize the value** of items being stole, while **not overfilling our backpack**



Value: 90
Weight: 9

Is this the **optimal** solution?

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

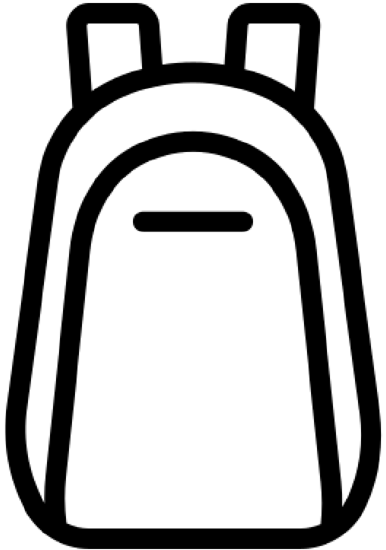
Value: 50
Weight: 8

Value: 5
Weight: 2

We are going to steal some items from the house and put them into our backpack

Each item has a weight, and a value

Goal: Steal items such that we **maximize the value** of items being stole, while **not overfilling our backpack**



Stuff our backpack with the most expensive items until we can’t fit anymore

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

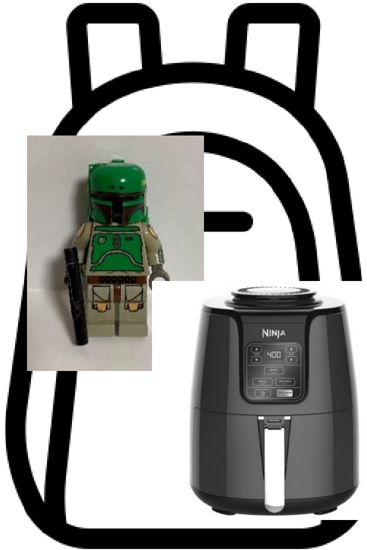
Value: 50
Weight: 8

Value: 5
Weight: 2

We are going to steal some items from the house and put them into our backpack

Each item has a weight, and a value

Goal: Steal items such that we **maximize the value** of items being stole, while **not overfilling our backpack**



Value: 55
Weight: 10

Is this the **optimal** solution?

How could we **prove** our algorithm wrong?

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

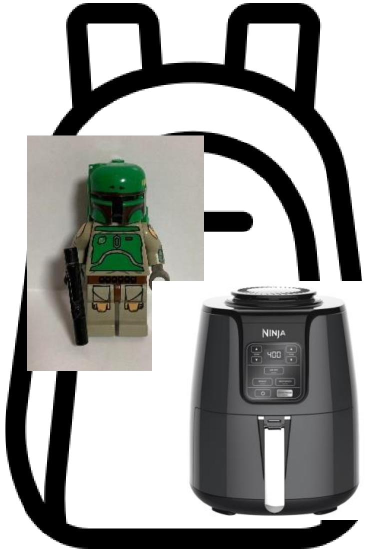
Value: 50
Weight: 8

Value: 5
Weight: 2

We are going to steal some items from the house and put them into our backpack

Each item has a weight, and a value

Goal: Steal items such that we **maximize the value** of items being stole, while **not overfilling our backpack**



Value: 55
Weight: 10

Is this the **optimal** solution?

NO

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

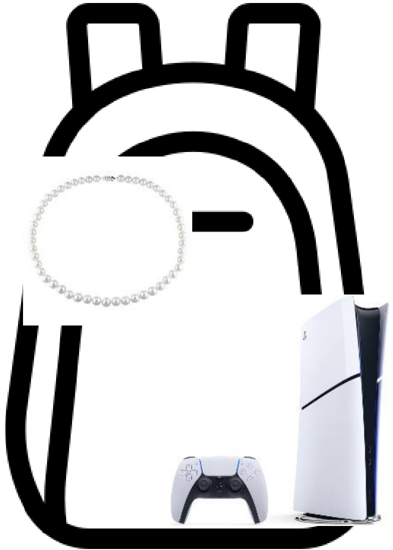
Value: 50
Weight: 8

Value: 5
Weight: 2

We are going to steal some items from the house and put them into our backpack

Each item has a weight, and a value

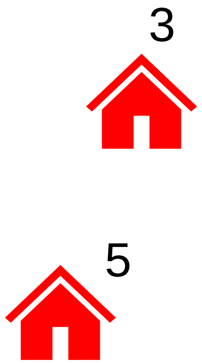
Goal: Steal items such that we **maximize the value** of items being stole, while **not overfilling our backpack**



Value: 70
Weight: 10

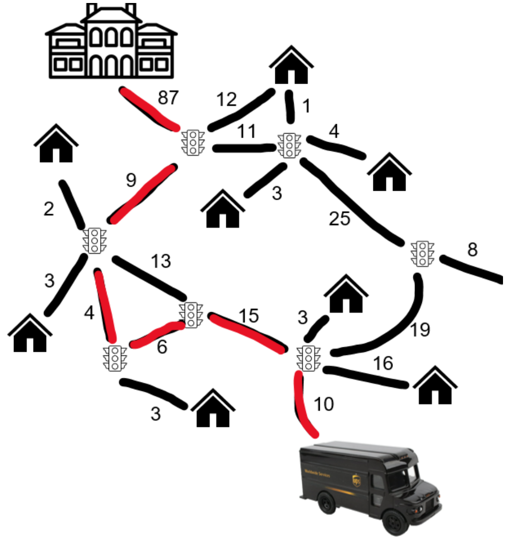
Our algorithm would never select this solution ☹

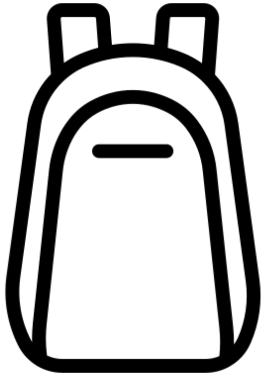
Takeaways:



Sometimes the most basic solution is **infeasible** or **inefficient**

Brute forcing is usually always infeasible for any arbitrary input





Sometimes our proposed algorithm won’t give us the correct answer

Takeaways:

3

5

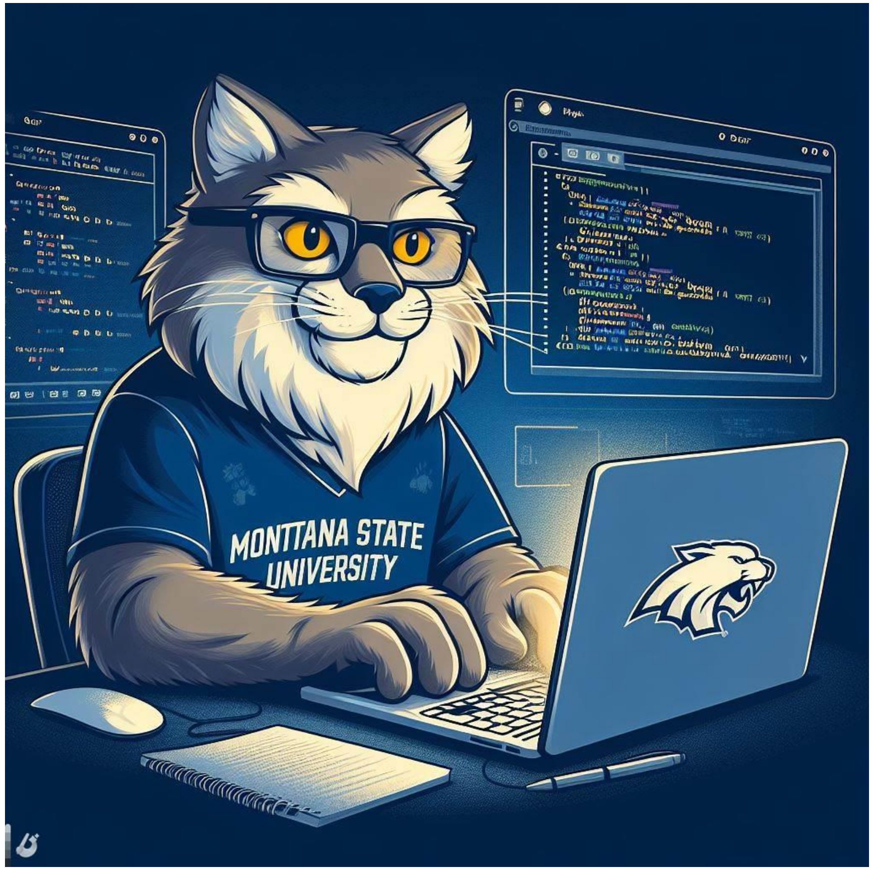
We need to produce more creative, efficient algorithms to solve problems

There are many ways to solve a problem, but some ways are better than others

give us the correct answer

Our Goals for this Semester

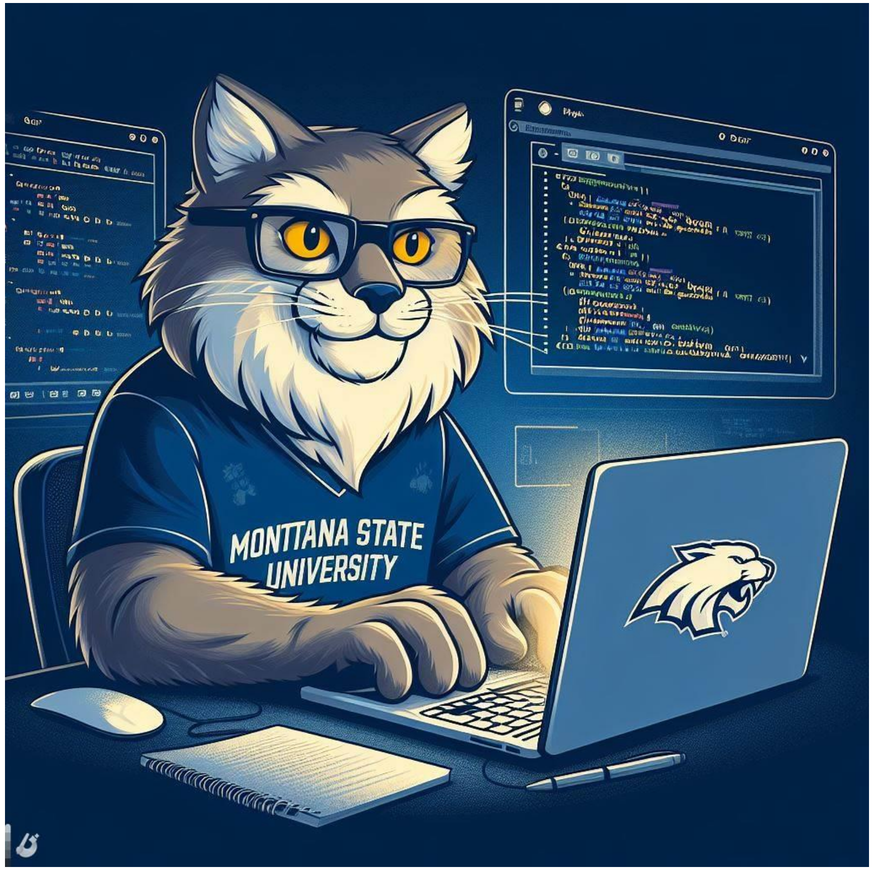
- Code (a lot) (in Java)



Our Goals for this Semester

- Code (a lot) (in Java)

- Learn about more **data structures** we can use in our programs (and tradeoffs)

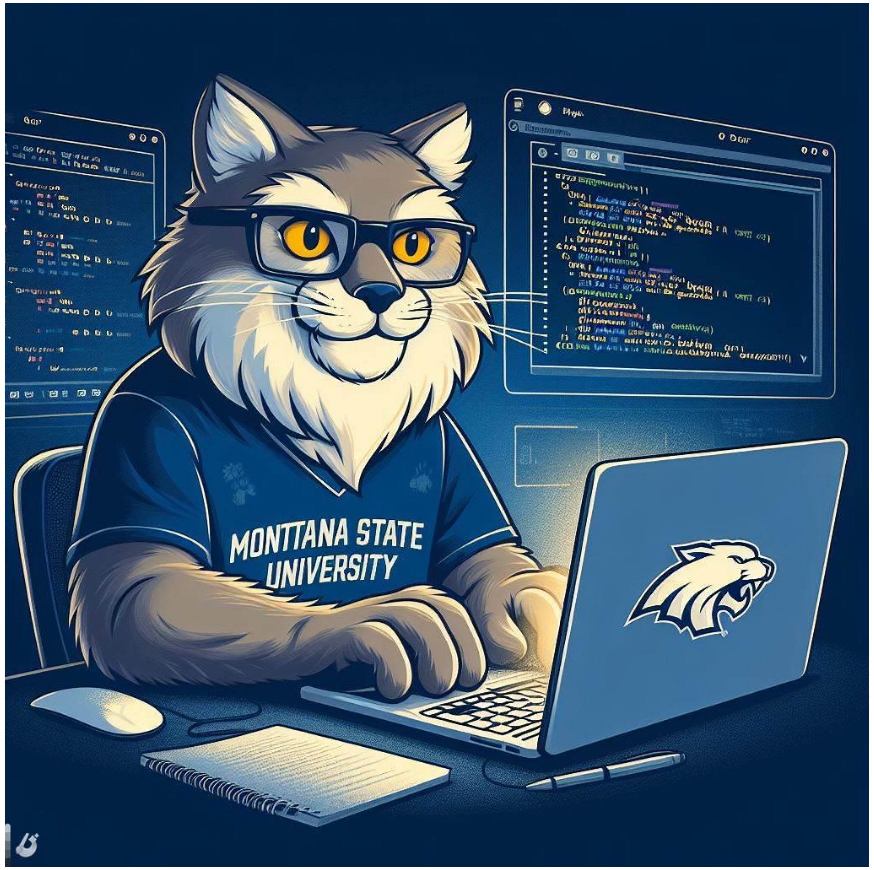


Our Goals for this Semester

- Code (a lot) (in Java)

- Learn about more **data structures** we can use in our programs (and tradeoffs)

- Learn about more strategies/types of **algorithms** to solve problems



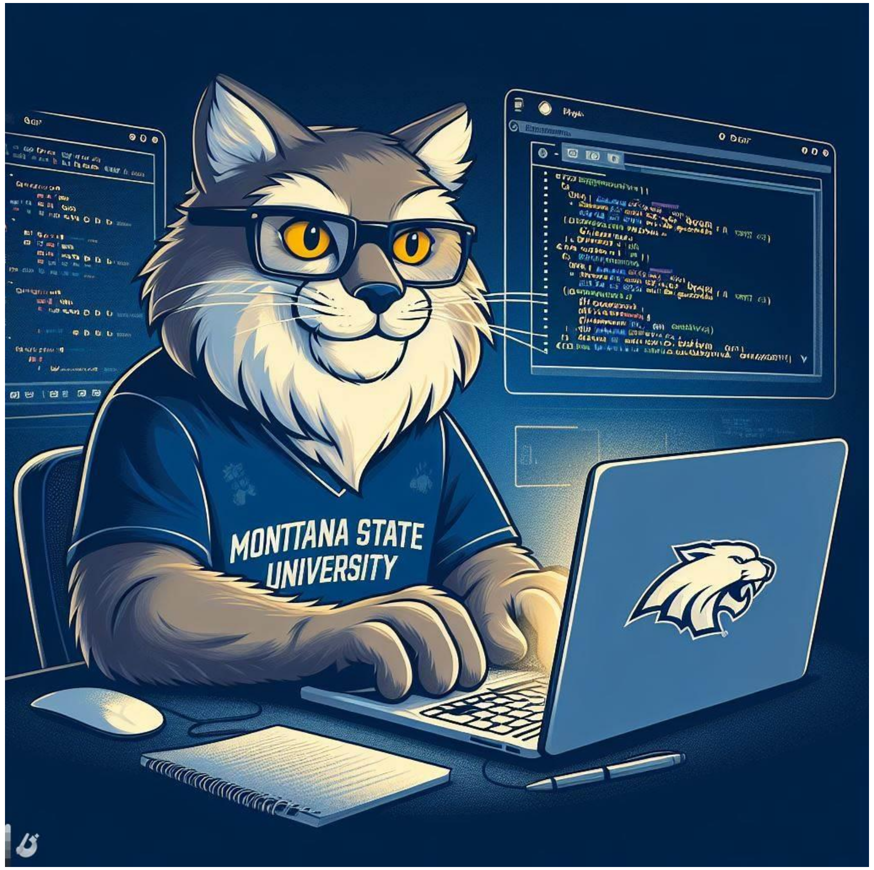
Our Goals for this Semester

- Code (a lot) (in Java)

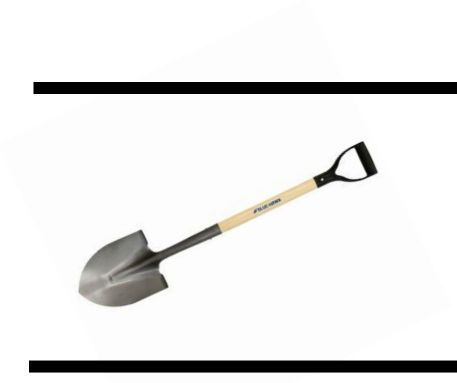
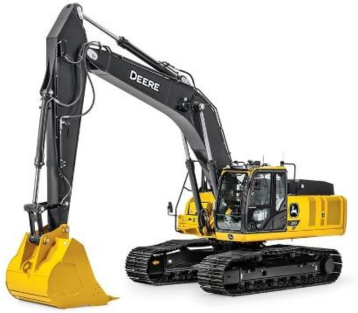
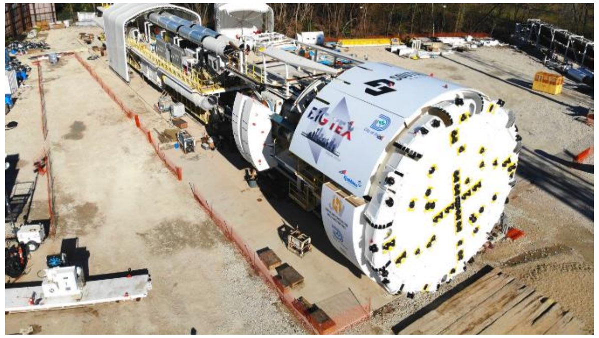
- Learn about more **data structures** we can use in our programs (and tradeoffs)

- Learn about more strategies/types of **algorithms** to solve problems

- Be able to formally detail the performance of an algorithm and identify factors limiting that performance



What do you need to dig a hole?

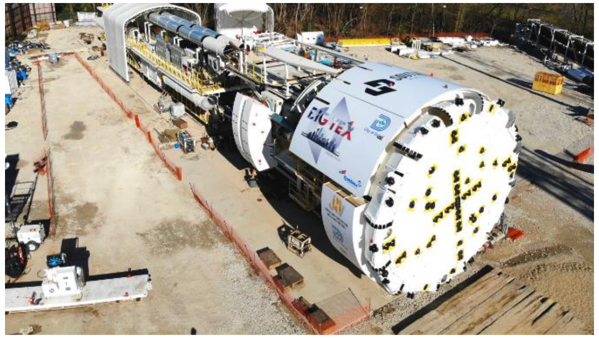
	Pros	Cons
		
		
		

What do you need to dig a hole?

	Pros	Cons
	• Cheap • Precise • No Training • Availability	• Slow • Labor
	• Fast • Labor	• Expensive • Training
	• Really good at digging	• Takes up a lot of garage space

Each tool has their pros, cons, and **tradeoffs**

What do you need to dig a hole?

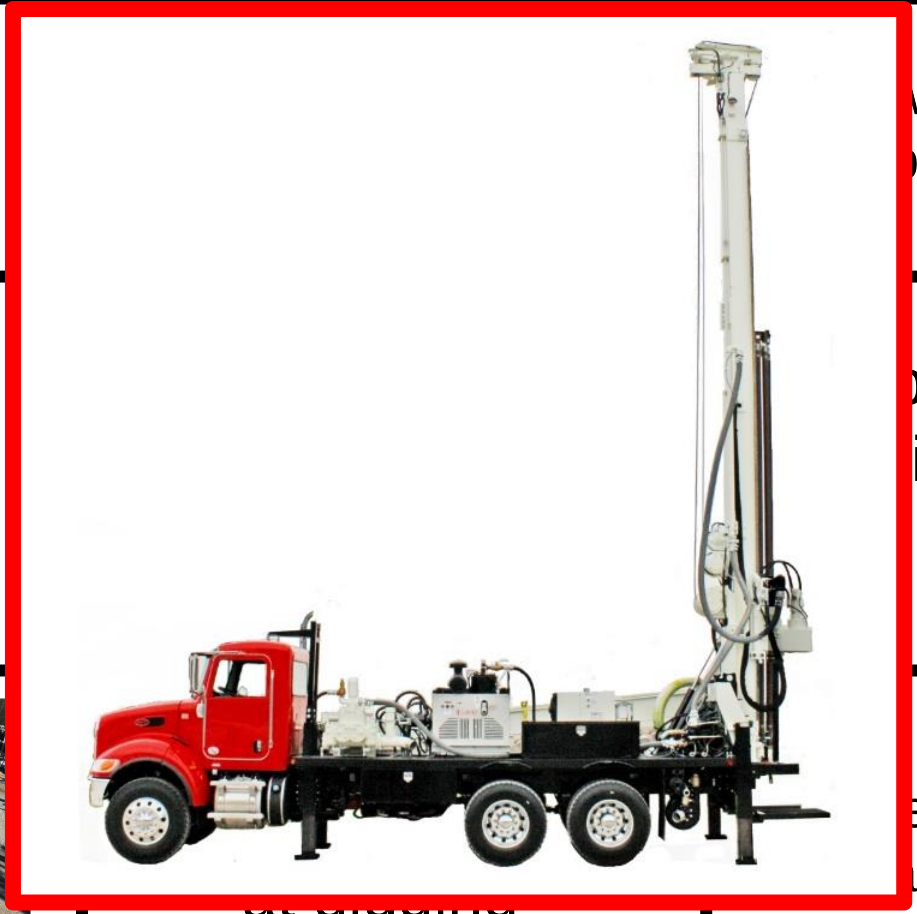
	Pros	**Cons**
	• Cheap • Precise • No Training • Availability	• Slow • Labor
	• Fast • Labor	• Expensive • Training
	• Really good at digging	• Takes up a lot of garage space

Best tool for the job?

Digging a Well for water

What do you need to dig a hole?

	Pros	Cons
		w or
		pensive ining
	at digging	es up a lot of ge space

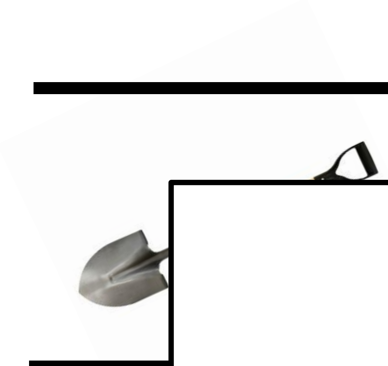


Best tool for the job?

Digging a Well for water

What do you need to dig a hole?

Pros **Cons**



at digging

es up a lot of
ge space

Best tool for the job?

a Well for

We can’t use the best tool for the job unless we know that tool exists!

What do you need to dig a hole?

of

at digging

What do you need to dig a hole?

at digging

We can’t use the best tool for the job unless we know how to use that tool

CSCI 232- Data Structures and Algorithms

“Tools”

- Arrays
- Linked Lists
- Stacks/Queues
- **Hash Tables**
- **Trees**
- **Graphs**

“Use of tools”

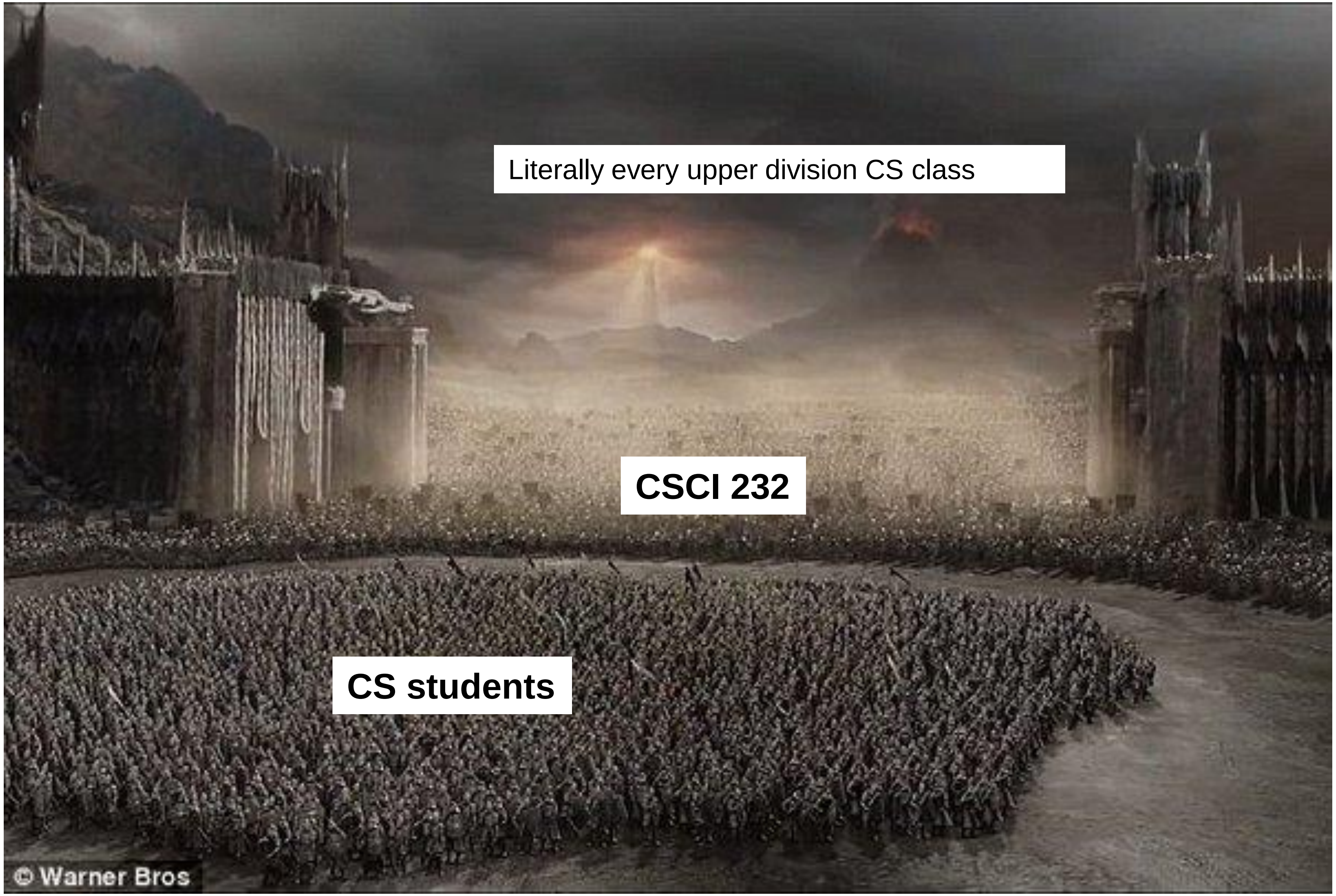
- Sorting
- Searching
- Routing
- Optimization

A **data structure** is a mechanism for storing and organizing data

An **algorithm** is a series of instructions to be followed to solve some problem

This class is **critical**

- Learn important set of tools to solve problems
- Become autonomous programmers, no more hand holding
- All other CS classes build on this*
- Job interview questions use stuff from this class


Literally every upper division CS class
CSCI 232
CS students
© Warner Bros

Reese Pearsall (pierce-all)

You can just call me "Reese" ☺

Second year Instructor @MSU
B.S & M.S @ MSU

Interests

- Cybersecurity
- Malware analysis and detection
- Cybercrime
- Computer Science Education

Hometown

- Billings, MT

Teaching

- CSCI 132
- CSCI 232

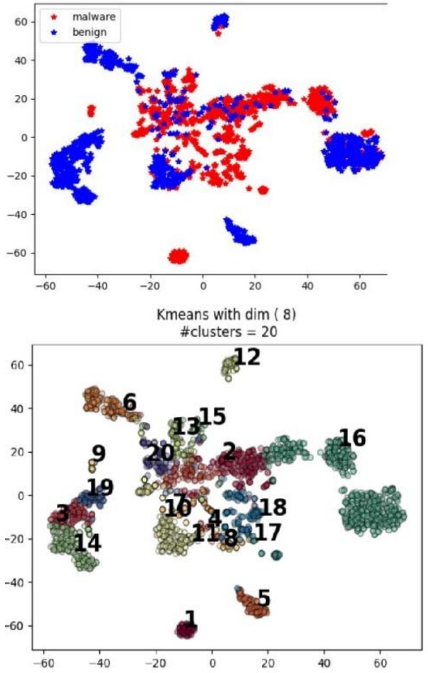
Experience

- Software Engineer and Tester, Techlink (Bozeman)
- Software Engineer, United States Air Force (Hill AFB, Utah)
- Cybersecurity Software Engineer, Hoplite Industries (Bozeman)
- Graduate Researcher, MSU (Bozeman)

Outside of academia

- Video games, New England Patriots, Fantasy Football, ~~TikTok~~, Movies, Dr Pepper, Memes, *The Bachelor*, Naps

Meatball



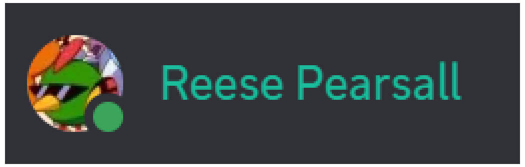
Contact

Email: reese.pearsall@montana.edu (I will respond as soon as I can)

Office Hours: Monday Wednesday 1:00 – 2:00 PM
Tuesday and Friday 12:10 – 1:00 PM

Office: Barnard Hall 361

When you email your professor at 2am and they respond within a minute


Reese Pearsall

I am also very responsive on Discord! (@reese_p)

College Student
@CollegeStudent

Emailing professors be like

Me: *polite greeting, multiple paragraphs, perfect grammar*

Professor: “sure” -sent from my iPhone

2:55 PM · 15 Oct 16

16K RETWEETS 31.5K LIKES

Course Logistics (Lecture)

<u>Class Meetings</u>

TR: 10:50 AM – 12:05 PM
Barnard Hall 103

- All lectures will be recorded and posted on the course website (coming to class is still a good idea)

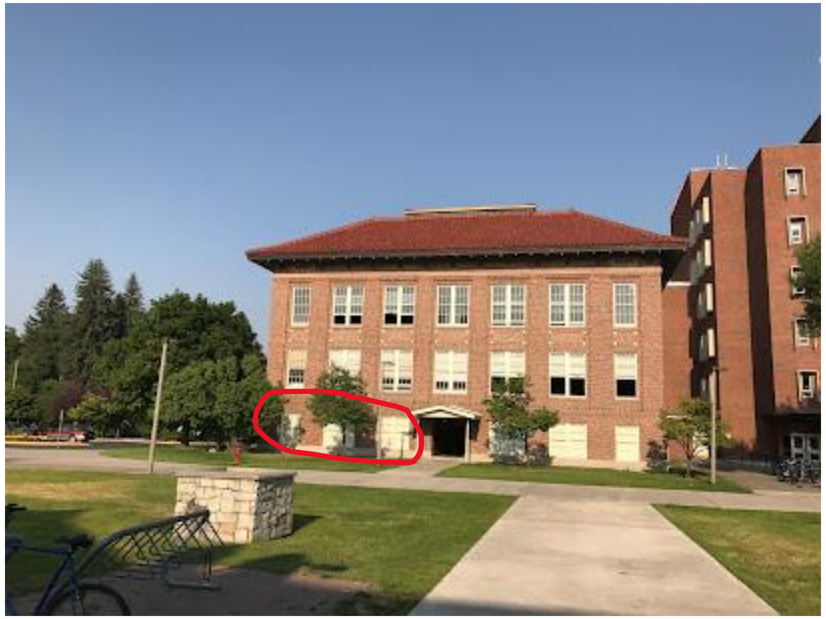
- We will be doing lots of live coding during lecture, so it might be helpful if you bring your own laptop to class (if you would like to code along)

- Please be respectful and considerate of your classmates siting around you

when I go to uni on 2h of sleep and the professor doesnt take attendance
ExtremeBlitz

Course Logistics (Lab)

•Section 003- Fridays 10:00 - 11:50 AM
•Section 004- Fridays 12:00 - 2:00 PM
•Section 005- Fridays 2:10 - 4:00 PM



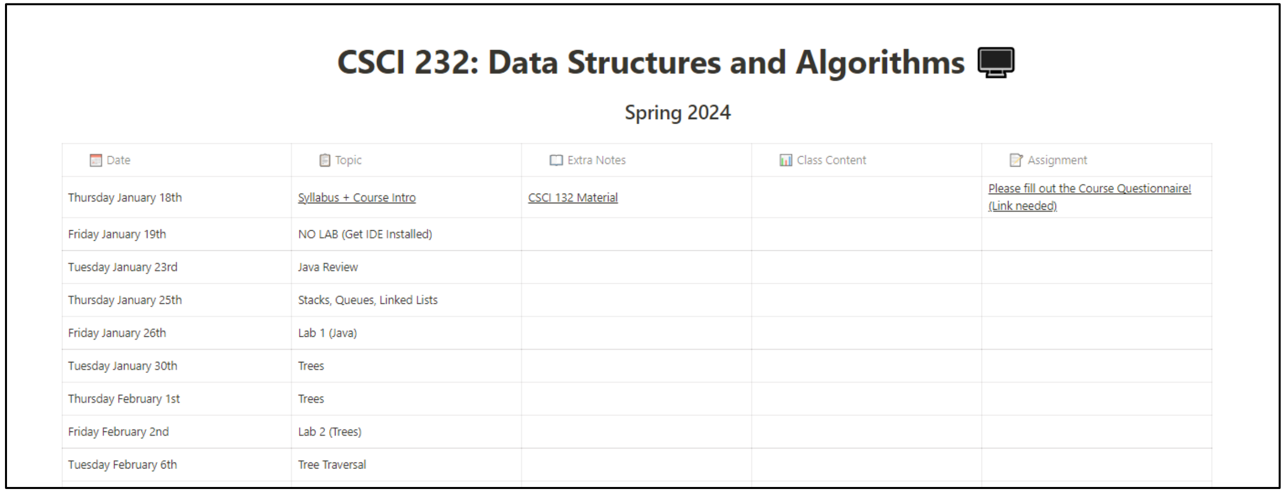
Locations: Roberts 111

- You can go to lab and get help from your TA and lab assistants
- Lab attendance is **optional**
- Lab assignments will be posted a few days before Fridays and can be completed from home.
- You can attend a different lab section earlier/later in the day if you would like

Course Logistics

You will be visiting this website a lot… be sure to bookmark it!

https://www.cs.montana.edu/pearsall/classes/spring2024/232/main.html



CSCI 232: Data Structures and Algorithms 🖥

Spring 2024

Date	Topic	Extra Notes	Class Content	Assignment
Thursday January 18th	Syllabus + Course Intro	CSCI 132 Material		Please fill out the Course Questionnaire! (Link needed)
Friday January 19th	NO LAB (Get IDE Installed)			
Tuesday January 23rd	Java Review			
Thursday January 25th	Stacks, Queues, Linked Lists			
Friday January 26th	Lab 1 (Java)			
Tuesday January 30th	Trees			
Thursday February 1st	Trees			
Friday February 2nd	Lab 2 (Trees)			
Tuesday February 6th	Tree Traversal			

Jan 18 - Syllabus

Jan 29

Feb12

Feb 22

Feb 29

March 12

March 26

April 18

April 30

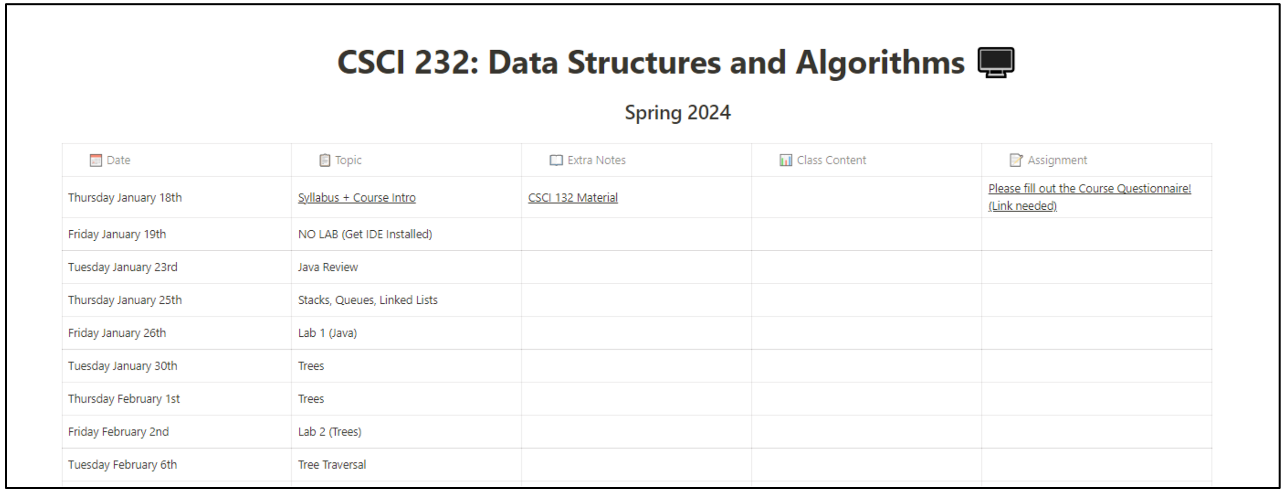
May 9 – Final Exam

Course Logistics

You will be visiting this website a lot… be sure to bookmark it!

https://www.cs.montana.edu/pearsall/classes/spring2024/232/main.html

(This schedule will change a lot)

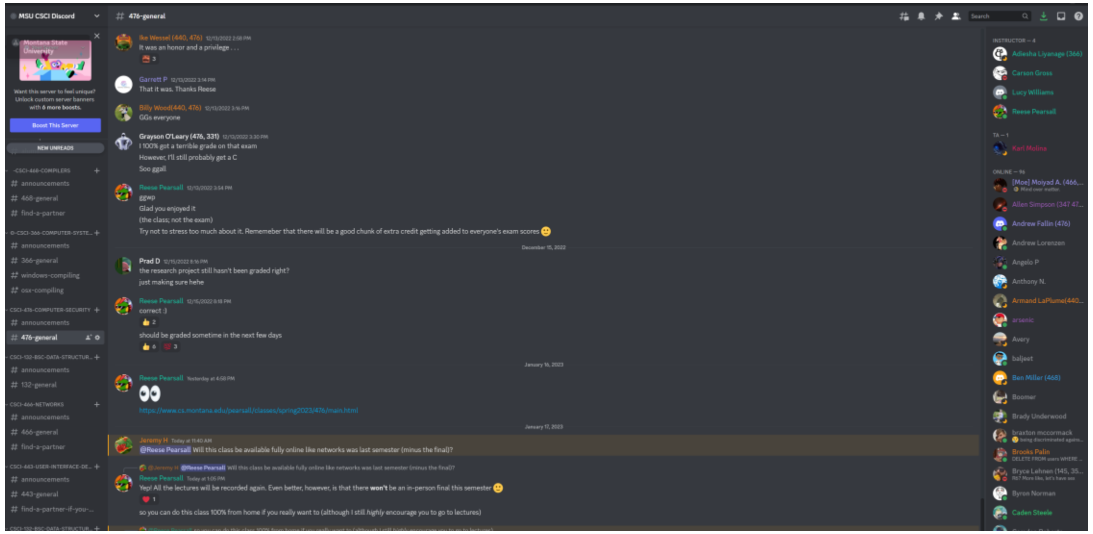


CSCI 232: Data Structures and Algorithms 🖥

Spring 2024

Date	Topic	Extra Notes	Class Content	Assignment
Thursday January 18th	Syllabus + Course Intro	CSCI 132 Material		Please fill out the Course Questionnaire! (Link needed)
Friday January 19th	NO LAB (Get IDE Installed)			
Tuesday January 23rd	Java Review			
Thursday January 25th	Stacks, Queues, Linked Lists			
Friday January 26th	Lab 1 (Java)			
Tuesday January 30th	Trees			
Thursday February 1st	Trees			
Friday February 2nd	Lab 2 (Trees)			
Tuesday February 6th	Tree Traversal			

You also will need to join our **Discord** server!



Ike Wessel (440, 476) 12/13/2022 2:58 PM
It was an honor and a privilege . . .
3

Garrett P 12/13/2022 3:14 PM
That it was. Thanks Reese

Billy Wood(440, 476) 12/13/2022 3:16 PM
GGs everyone

Grayson O'Leary (476, 331) 12/13/2022 3:30 PM
I 100% got a terrible grade on that exam
However, I'll still probably get a C
Soo ggall

Reese Pearsall 12/13/2022 3:54 PM
ggwp
Glad you enjoyed it
(the class; not the exam)
Try not to stress too much about it. Rememeber that there will be a good chunk of extra credit getting added to everyone's exam scores 🙂

December 15, 2022

Prad D 12/15/2022 8:16 PM
the research project still hasn't been graded right?
just making sure hehe

Reese Pearsall 12/15/2022 8:18 PM
correct :)
👍 2
should be graded sometime in the next few days
👍 6 💯 3

January 16, 2023

Reese Pearsall Yesterday at 4:58 PM
👀
https://www.cs.montana.edu/pearsall/classes/spring2023/476/main.html

January 17, 2023

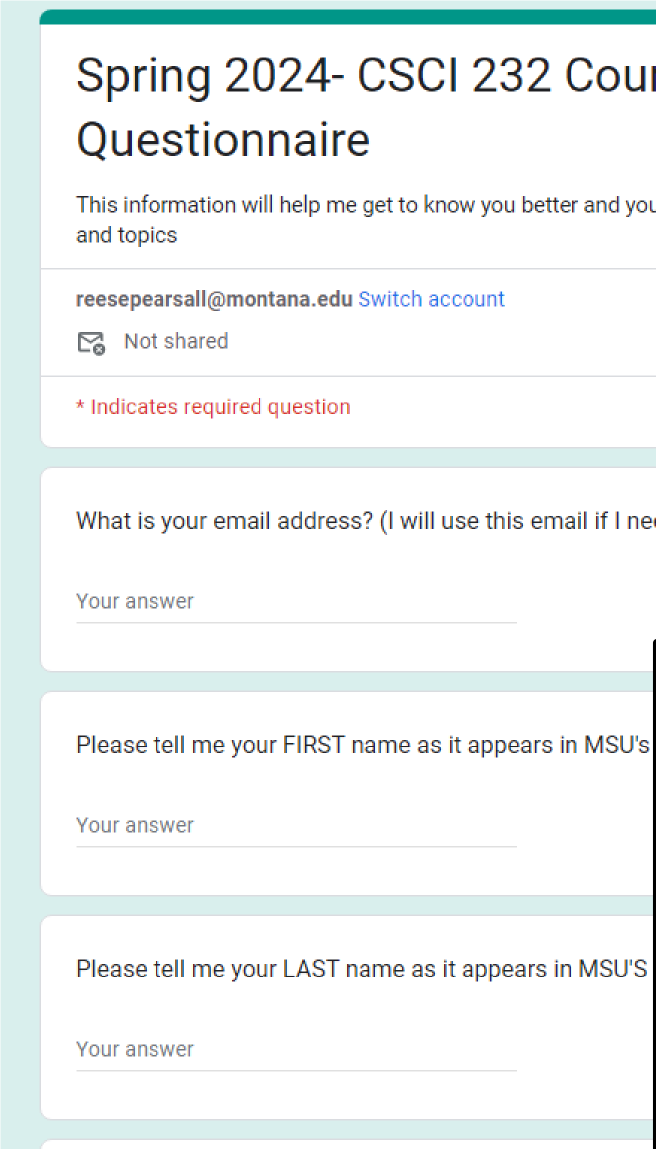
Jeremy H Today at 11:40 AM
@Reese Pearsall Will this class be available fully online like networks was last semester (minus the final)?

@Jeremy H @Reese Pearsall Will this class be available fully online like networks was last semester (minus the final)?
Reese Pearsall Today at 1:05 PM
Yep! All the lectures will be recorded again. Even better, however, is that there **won't** be an in-person final this semester 🙂
❤️ 1
so you can do this class 100% from home if you really want to (although I still *highly* encourage you to go to lectures)

Get 232 notifications
by typing `!join-232`

Course Questionnaire

Please take some time this week to fill out the course questionnaire ☺



Spring 2024- CSCI 232 Course Questionnaire

This information will help me get to know you better and your experience with various tools and topics

reesepearsall@montana.edu Switch account

Not shared

* Indicates required question

What is your email address? (I will use this email if I need to contact you) *

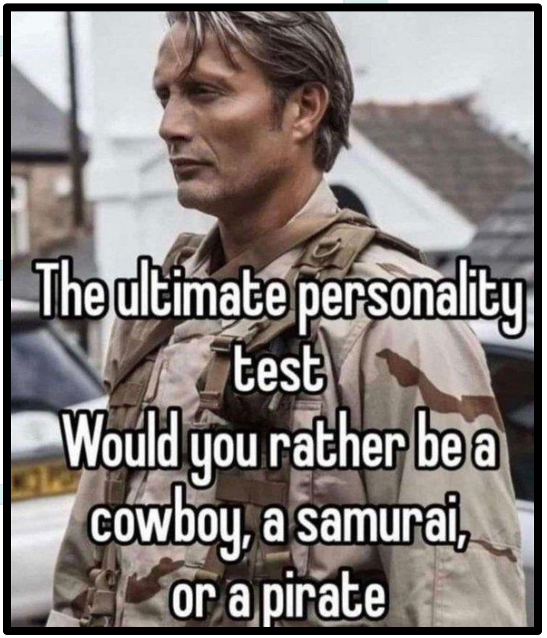
Your answer

Please tell me your FIRST name as it appears in MSU's

Your answer

Please tell me your LAST name as it appears in MSU'S

Your answer


The ultimate personality test
Would you rather be a cowboy, a samurai, or a pirate

Prerequisites

- CSCI 132- Basic Data Structures and Algorithms (Required)
- ~~CSCI 246- Discrete Structures (Recommended)~~

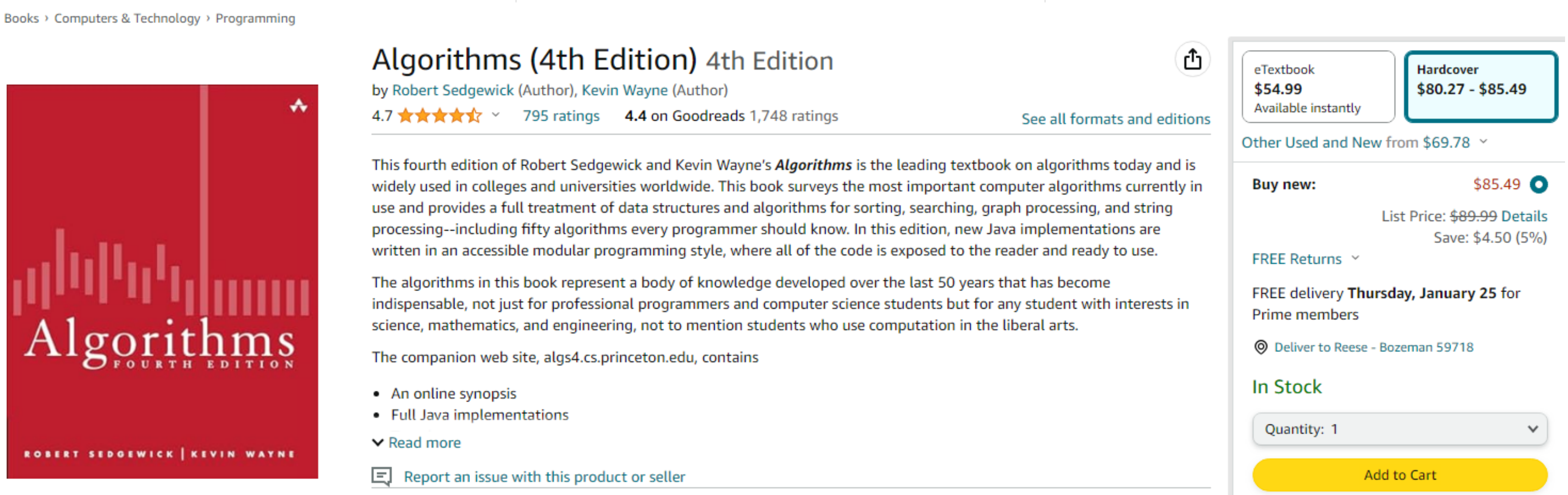
*You will be totally fine if you have taken 246

Before taking this class, you should feel comfortable basic Java programming, be comfortable using the following data structures: arrays, linked lists, stacks, queues, be comfortable with basic recursion, and how to analyze an algorithm using big-O notation

(If you are not familiar with any of this stuff, you should take some time to review it this week. My CSCI 132 course is available and may be helpful)

Textbook

- (Optional) Algorithms (4th Edition) by Sedgewick and Wayne.



Books › Computers & Technology › Programming

Algorithms (4th Edition) 4th Edition

by Robert Sedgewick (Author), Kevin Wayne (Author)

4.7 ★★★★☆ 795 ratings 4.4 on Goodreads 1,748 ratings

See all formats and editions

This fourth edition of Robert Sedgewick and Kevin Wayne's ***Algorithms*** is the leading textbook on algorithms today and is widely used in colleges and universities worldwide. This book surveys the most important computer algorithms currently in use and provides a full treatment of data structures and algorithms for sorting, searching, graph processing, and string processing--including fifty algorithms every programmer should know. In this edition, new Java implementations are written in an accessible modular programming style, where all of the code is exposed to the reader and ready to use.

The algorithms in this book represent a body of knowledge developed over the last 50 years that has become indispensable, not just for professional programmers and computer science students but for any student with interests in science, mathematics, and engineering, not to mention students who use computation in the liberal arts.

The companion web site, algs4.cs.princeton.edu, contains

- An online synopsis
- Full Java implementations

Read more

Report an issue with this product or seller

eTextbook
$54.99
Available instantly

Hardcover
$80.27 - $85.49

Other Used and New from $69.78

Buy new: $85.49 ●

List Price: ~~$89.99~~ Details
Save: $4.50 (5%)

FREE Returns

FREE delivery **Thursday, January 25** for Prime members

Deliver to Reese - Bozeman 59718

In Stock

Quantity: 1

Add to Cart

@bejewelledbud

Can you guys please recommend books that made you cry?

Frease
@FreaseDaddy

Data Structures and Algorithms in Java (2nd Edition) 2nd Edition

by Robert Lafore (Author)

★★★★☆ 114 customer reviews

Look inside

Kindle
$29.80

Hardcover
$33.89 - $45.04

Paperback
$23.39 - $27.18

Other Se
See all 6 versi

○ Buy used

● Buy new

In Stock.

unfortunately, a very relatable meme

This textbook is **not** required
(but it does have tons of great stuff!!)

Grading

- 30% - Labs (12 @ ~3% each)
- 40% - Programs (4 @ 10% each)
- 15% - Midterm
- 15% - Final Exam

Grading

Labs (30%)

- Shorter, weekly assignments.
- Can generally be finished within 1-2.5 hours
- Due on Friday nights @ 11:59 PM
- I will post the labs a few days ahead of time
- You should be able to finish within your 2hr lab time
- I will drop your lowest lab grade at the end of the semester
- Individual submissions

Grading

Programs (40%)

- Longer, more complicated programming assignments
- Will likely take 2+ hours to complete
- You will always have 2-3 weeks to complete them
- Much higher stakes, make sure you give yourself plenty of time to complete them
- You can get help from your TA during lab time, or office hours, or from Reese, or on Discord
- You are allowed to work with 1 partner

Grading

Exams (Midterm and Final) (30%)

Midterm: Thursday March 21st
Final: Thursday May 9th

- Exams consist of short answer, multiple choice, true/false, and some small coding problems
- You are allowed to use your laptop and any notes

Grading Deductions

- If you submit late, but you are within < 24 of the original. You will face a -25% penalty
- If you submit late, but you are within < 48 of the original. You will face a -50% penalty

Any assignment submitted 48+ hours after the deadline will **not** be accepted

You must submit code that **compiles**. Code that does not compile will receive an automatic 0%.

If your code compiles and runs, but doesn’t work, or has **runtime errors** later on, that is ok.

Your TA or I should not need to fix your code in order for it to compile and run

Grading Scale

- 93+: A
- 90+: A-
- 87+: B+
- 83+: B
- 80+: B-
- 77+: C+
- 73+: C
- 70+: C-
- 67+: D+
- 63: D
- 60: D-

At the end of the semester, if you are within 1% of the next letter grade, I will bump you up

I will not curve exams or final grades unless it is needed

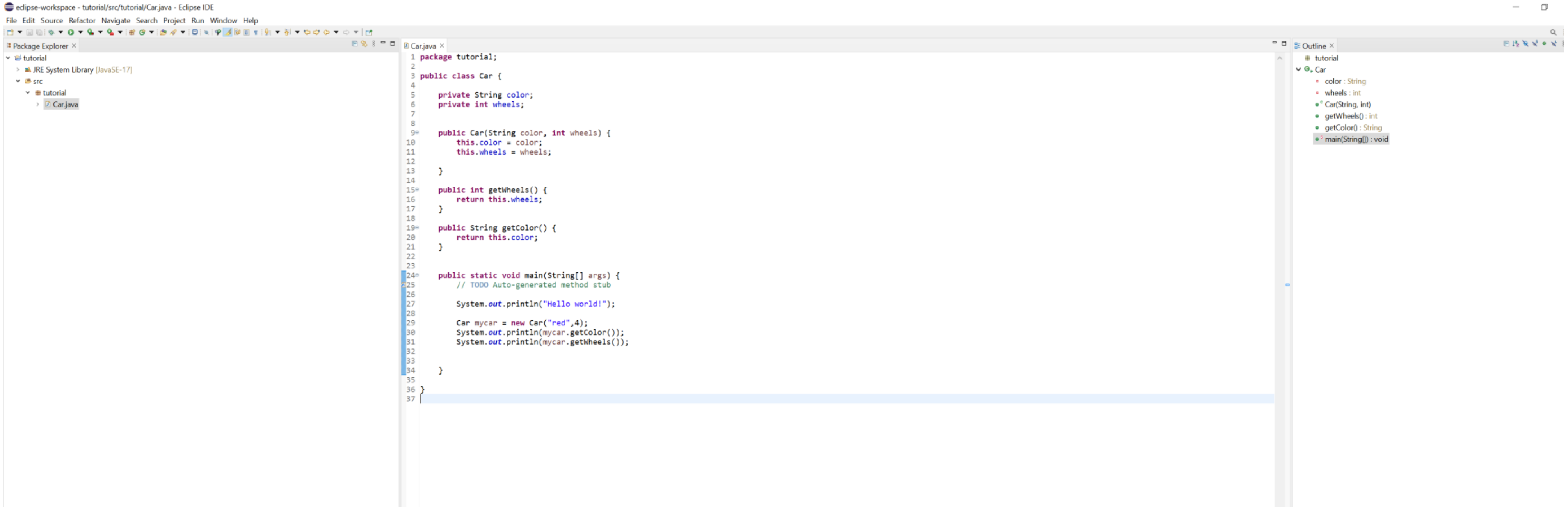
juju 💰
@ihyjuju

in college you gotta get over L’s real quick because the next one is due at 11:59

IDE

You will need to download an IDE that you can write Java programs in

- Eclipse (I will use this one)
- Netbeans
- IntelliJ



```java
package tutorial;

public class Car {

    private String color;
    private int wheels;


    public Car(String color, int wheels) {
        this.color = color;
        this.wheels = wheels;

    }

    public int getWheels() {
        return this.wheels;
    }

    public String getColor() {
        return this.color;
    }


    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.out.println("Hello world!");

        Car mycar = new Car("red",4);
        System.out.println(mycar.getColor());
        System.out.println(mycar.getWheels());


    }

}
```

Academic Misconduct

Plagiarism and cheating is very not cool

You are **not** allowed to submit something that is not your own, and you are **not** allowed to steal solutions from another person and modify it

I have a Chegg and Course Hero membership. **Don’t try it**

Do not use any tools or AI that will write code for you

Using small snippets of code from the internet is acceptable *(but should not be needed).* If you do use a small snippet of code from the internet, you should leave a reference as a comment in your code

Collaboration Policy

All labs will be individual submissions.
For programs, you are allowed to work with **one** partner.

When it comes to labs, you *may*

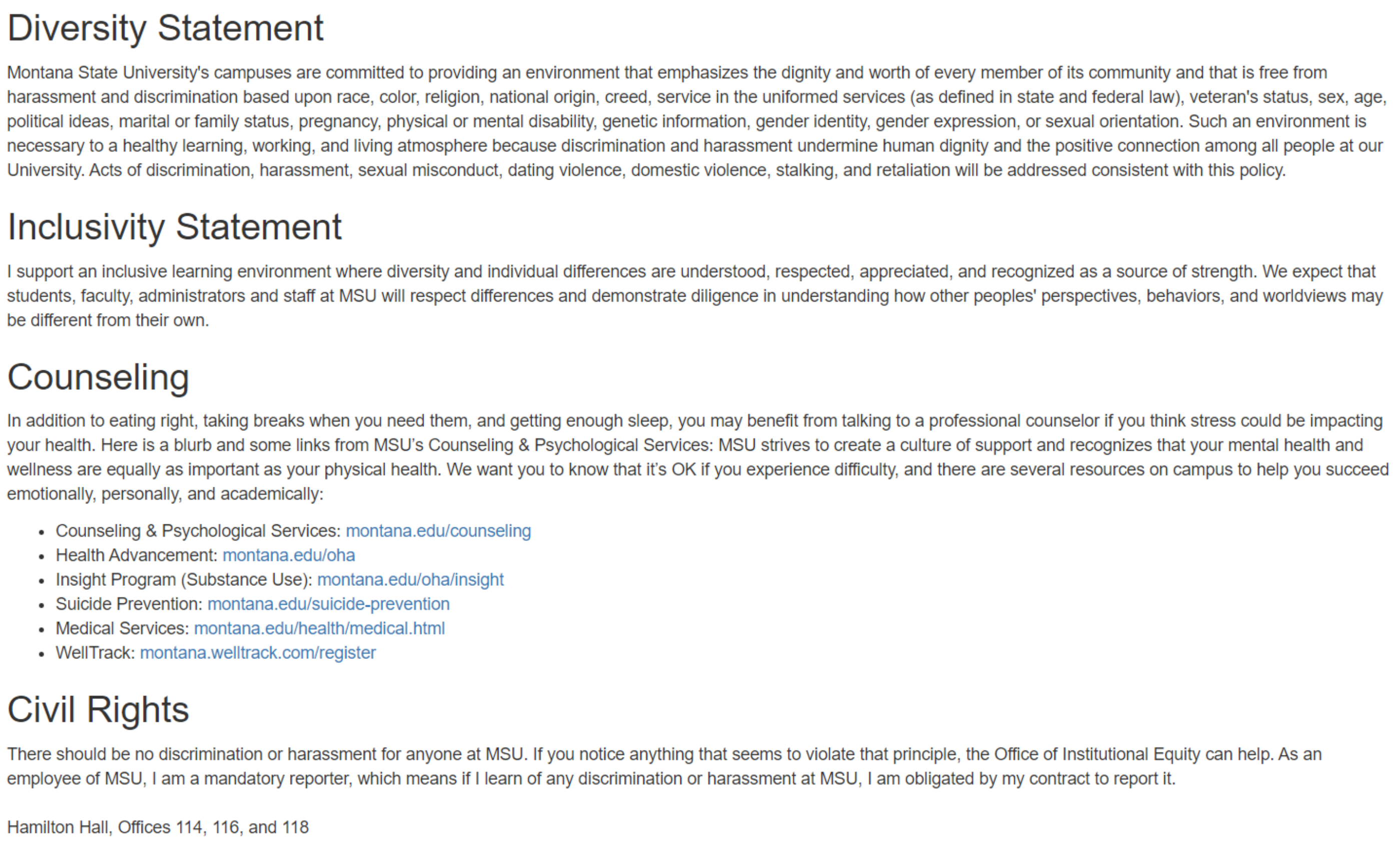
- Share ideas with other students in the class.
- Work together on labs in the same physical location.
- Help other students troubleshoot problems.
- Give hints or provide textbook page numbers/slide numbers to students seeking help

You may *NOT*

- Share your code and solutions directly with other students.
- Submit solutions that you did not write.
- Modify another student's solution and claim it as your own.
- Share your report or solutions directly on Discord

Additional MSU Resources:

https://www.cs.montana.edu/pearsall/classes/msu_resources.html



Diversity Statement

Montana State University's campuses are committed to providing an environment that emphasizes the dignity and worth of every member of its community and that is free from harassment and discrimination based upon race, color, religion, national origin, creed, service in the uniformed services (as defined in state and federal law), veteran's status, sex, age, political ideas, marital or family status, pregnancy, physical or mental disability, genetic information, gender identity, gender expression, or sexual orientation. Such an environment is necessary to a healthy learning, working, and living atmosphere because discrimination and harassment undermine human dignity and the positive connection among all people at our University. Acts of discrimination, harassment, sexual misconduct, dating violence, domestic violence, stalking, and retaliation will be addressed consistent with this policy.

Inclusivity Statement

I support an inclusive learning environment where diversity and individual differences are understood, respected, appreciated, and recognized as a source of strength. We expect that students, faculty, administrators and staff at MSU will respect differences and demonstrate diligence in understanding how other peoples' perspectives, behaviors, and worldviews may be different from their own.

Counseling

In addition to eating right, taking breaks when you need them, and getting enough sleep, you may benefit from talking to a professional counselor if you think stress could be impacting your health. Here is a blurb and some links from MSU's Counseling & Psychological Services: MSU strives to create a culture of support and recognizes that your mental health and wellness are equally as important as your physical health. We want you to know that it's OK if you experience difficulty, and there are several resources on campus to help you succeed emotionally, personally, and academically:

- Counseling & Psychological Services: montana.edu/counseling
- Health Advancement: montana.edu/oha
- Insight Program (Substance Use): montana.edu/oha/insight
- Suicide Prevention: montana.edu/suicide-prevention
- Medical Services: montana.edu/health/medical.html
- WellTrack: montana.welltrack.com/register

Civil Rights

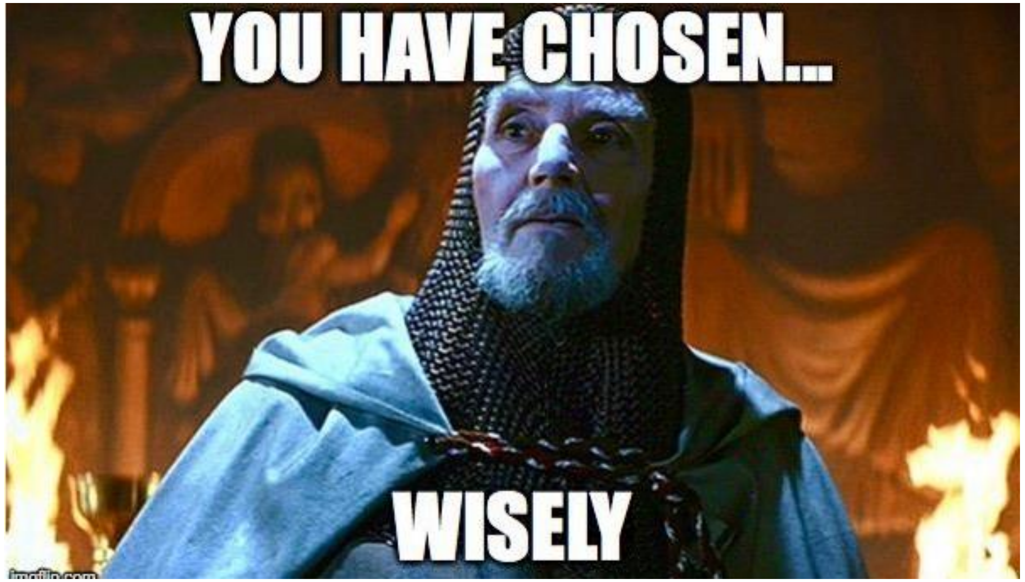
There should be no discrimination or harassment for anyone at MSU. If you notice anything that seems to violate that principle, the Office of Institutional Equity can help. As an employee of MSU, I am a mandatory reporter, which means if I learn of any discrimination or harassment at MSU, I am obligated by my contract to report it.

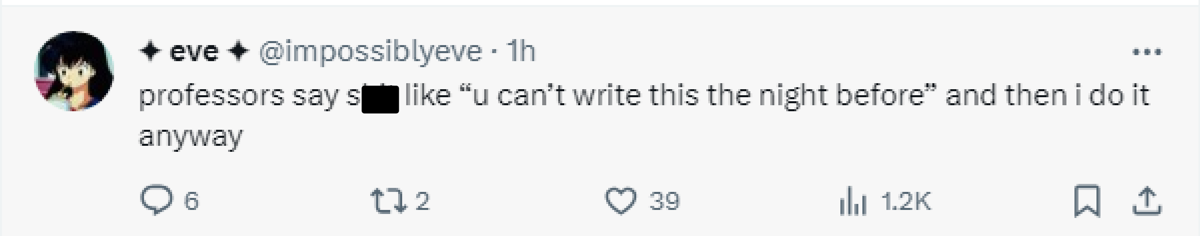
Hamilton Hall, Offices 114, 116, and 118

“Not everyone can become a great artist, but a great artist can come from anywhere”

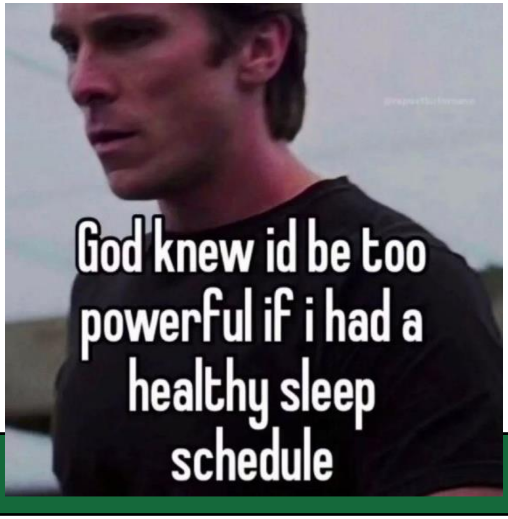
How to do well in this class

- **Get help when you need it**
- Get started on assignments early (especially programs)!
- Come to class and office hours
- Take care of yourself




✦ eve ✦ @impossiblyeve · 1h

professors say s[redacted] like "u can't write this the night before" and then i do it anyway


God knew id be too powerful if i had a healthy sleep schedule

How to do well in this class

- **Get help when you need it**

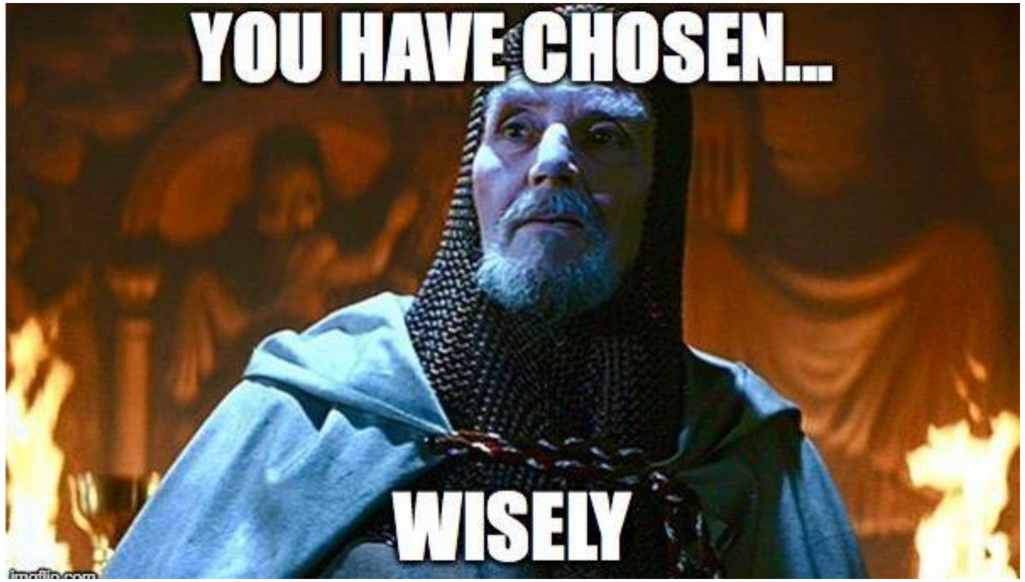
- Get started on assignments early (especially programs)!

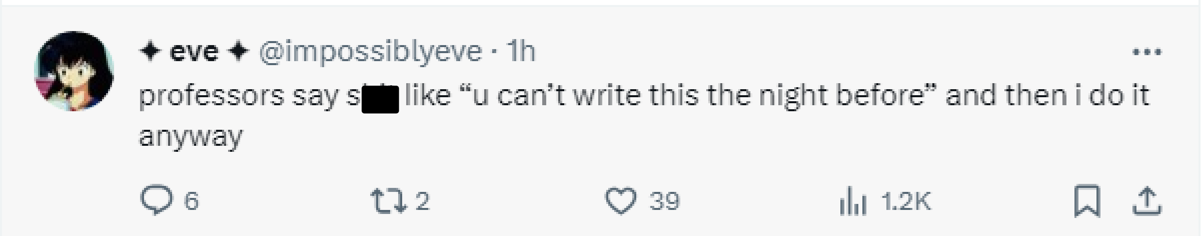
- Come to class and office hours

- Take care of yourself

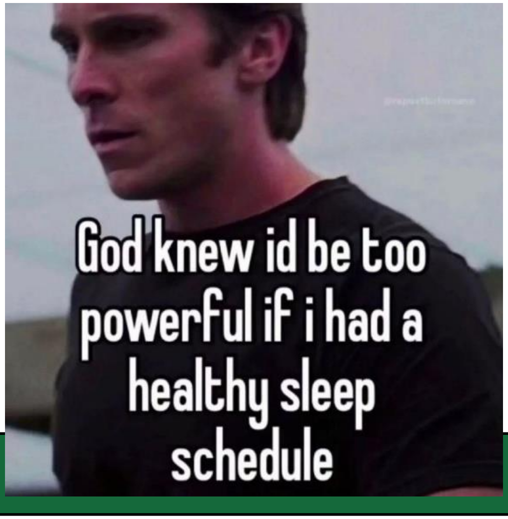
- **Try to have fun**

I am here for you, and I am willing to do whatever it takes to help you succeed!




✦ eve ✦ @impossiblyeve · 1h

professors say s[redacted] like “u can’t write this the night before” and then i do it anyway


God knew id be too powerful if i had a healthy sleep schedule

Questions?