CSCI232:

~~Basic~~ Data Structures and Algorithms

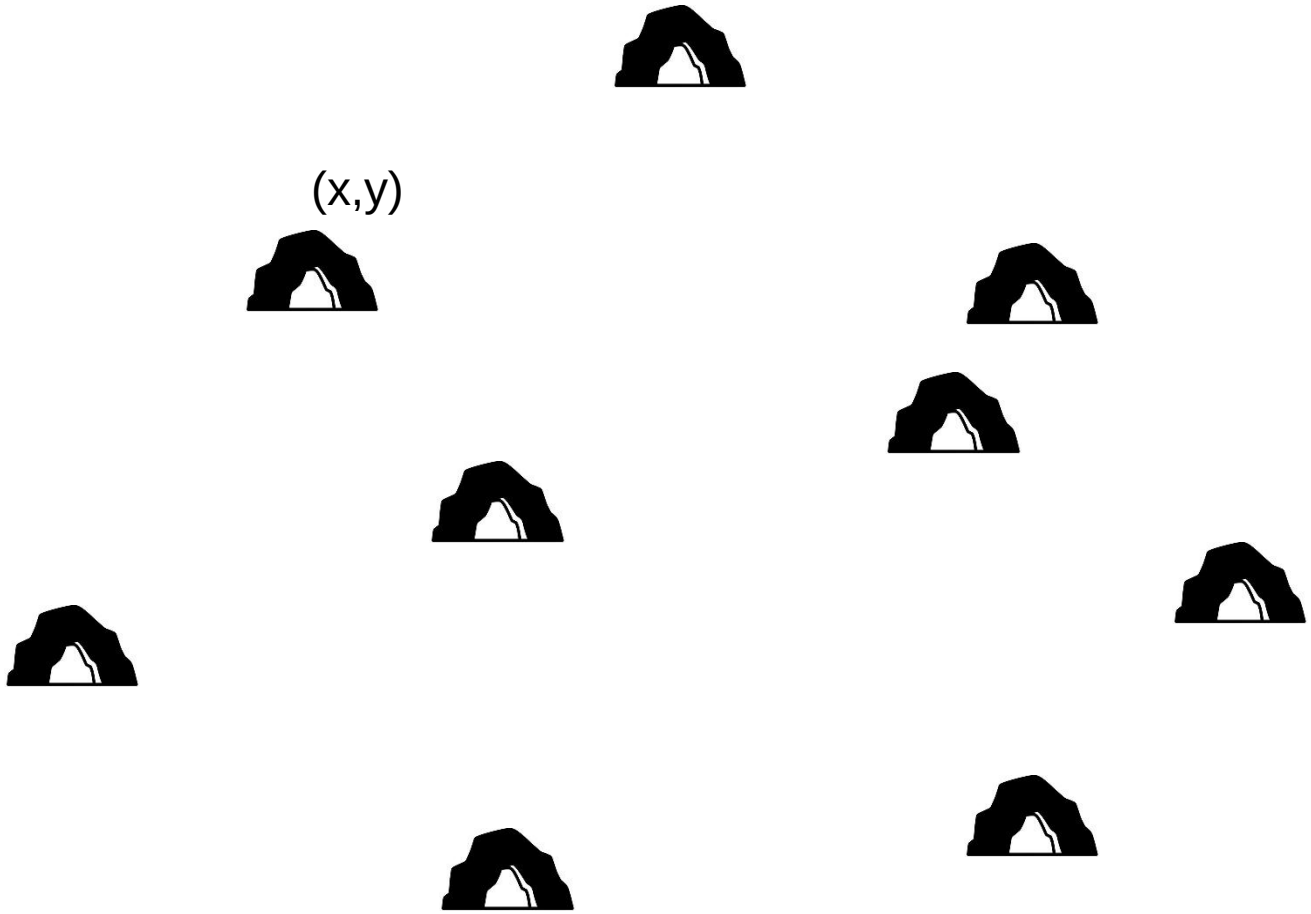
Course Intro, Syllabus, and Logistics

Reese Pearsall

Summer 2025

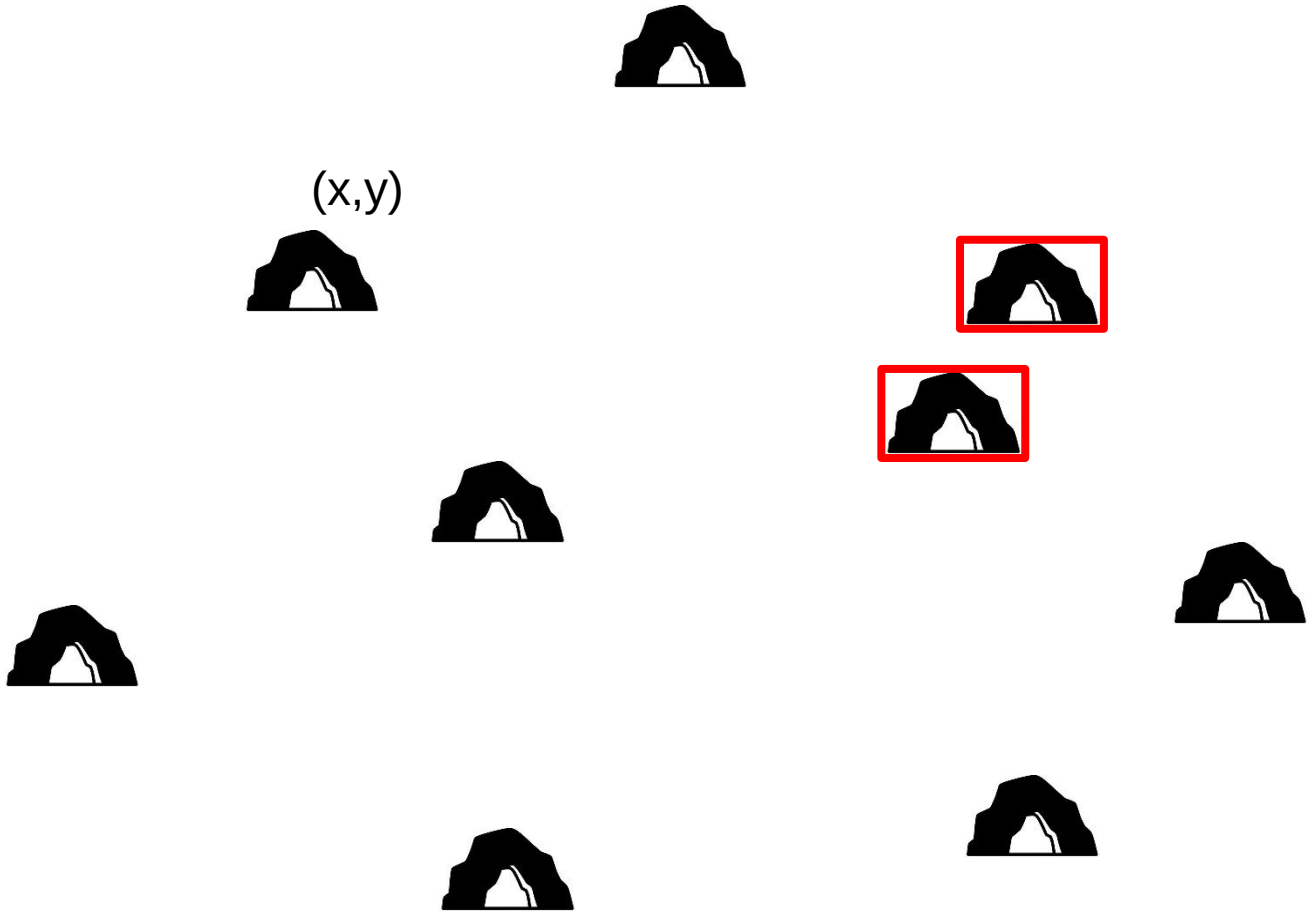
To start things off, we are going to take a brief look at some of problems we are going to tackle in CSCI 232

We are going to find a **cave** and go **treasure hunting**



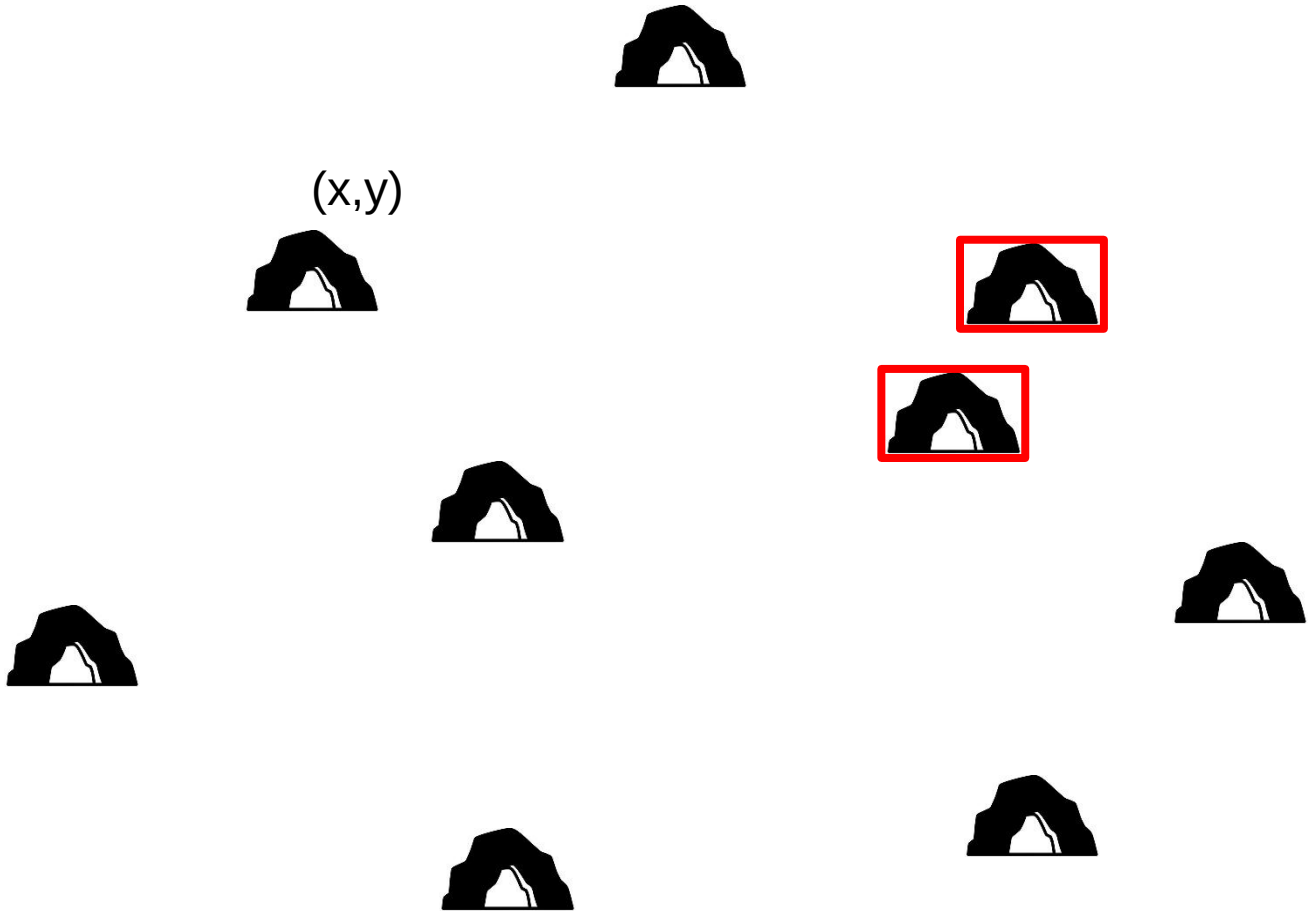
Given *n* cave entrances, where each cave entrance has an x,y coordinate, find the pair of caves with the smallest distance between them

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



Given *n* cave entrances, where each cave entrance has an x,y coordinate, find the pair of caves with the smallest distance between them

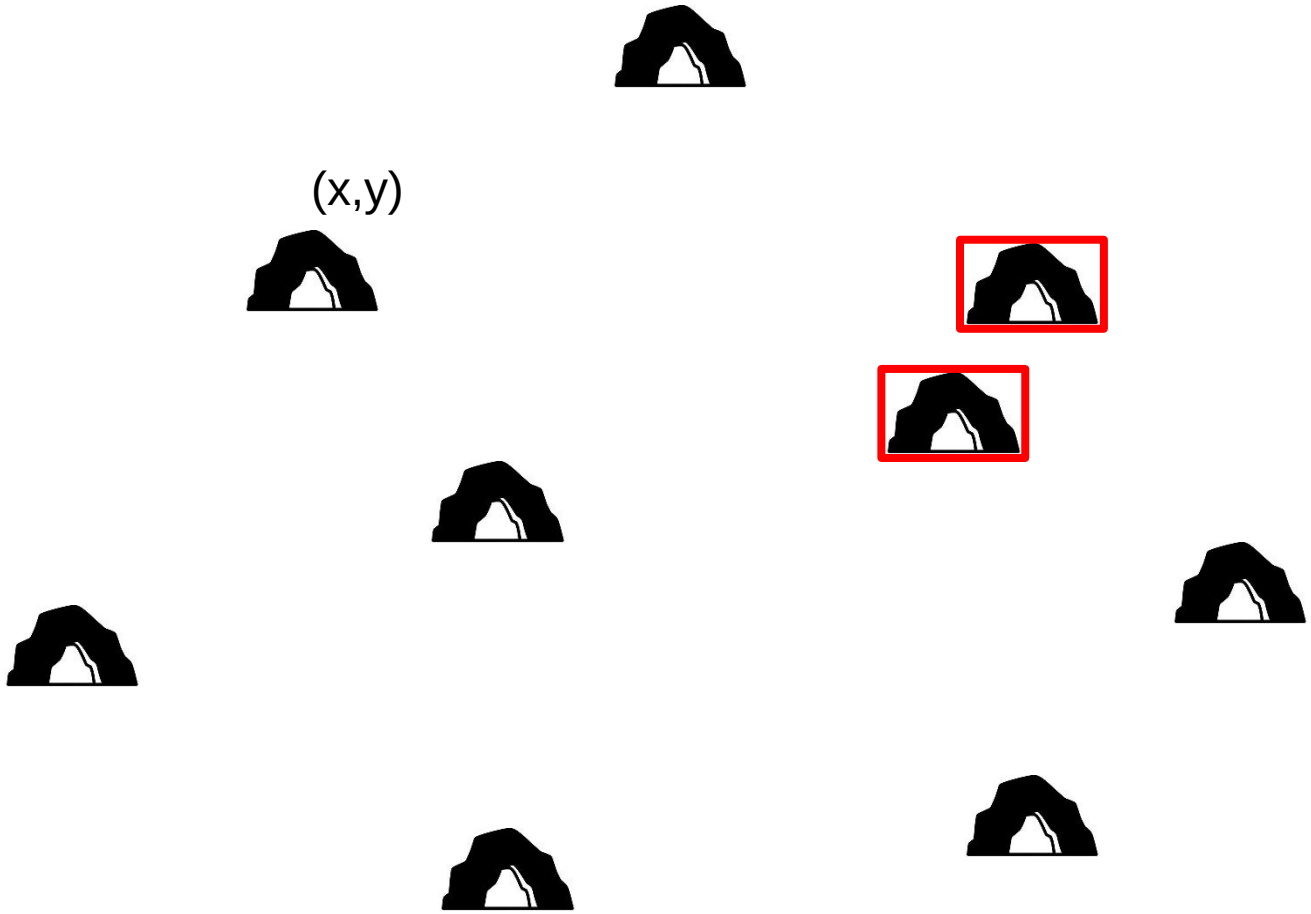
(You can assume no caves have the same x or y values)



Given *n* cave entrances, where each cave entrance has an x,y coordinate, find the pair of caves with the smallest distance between them

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

Algorithm ?

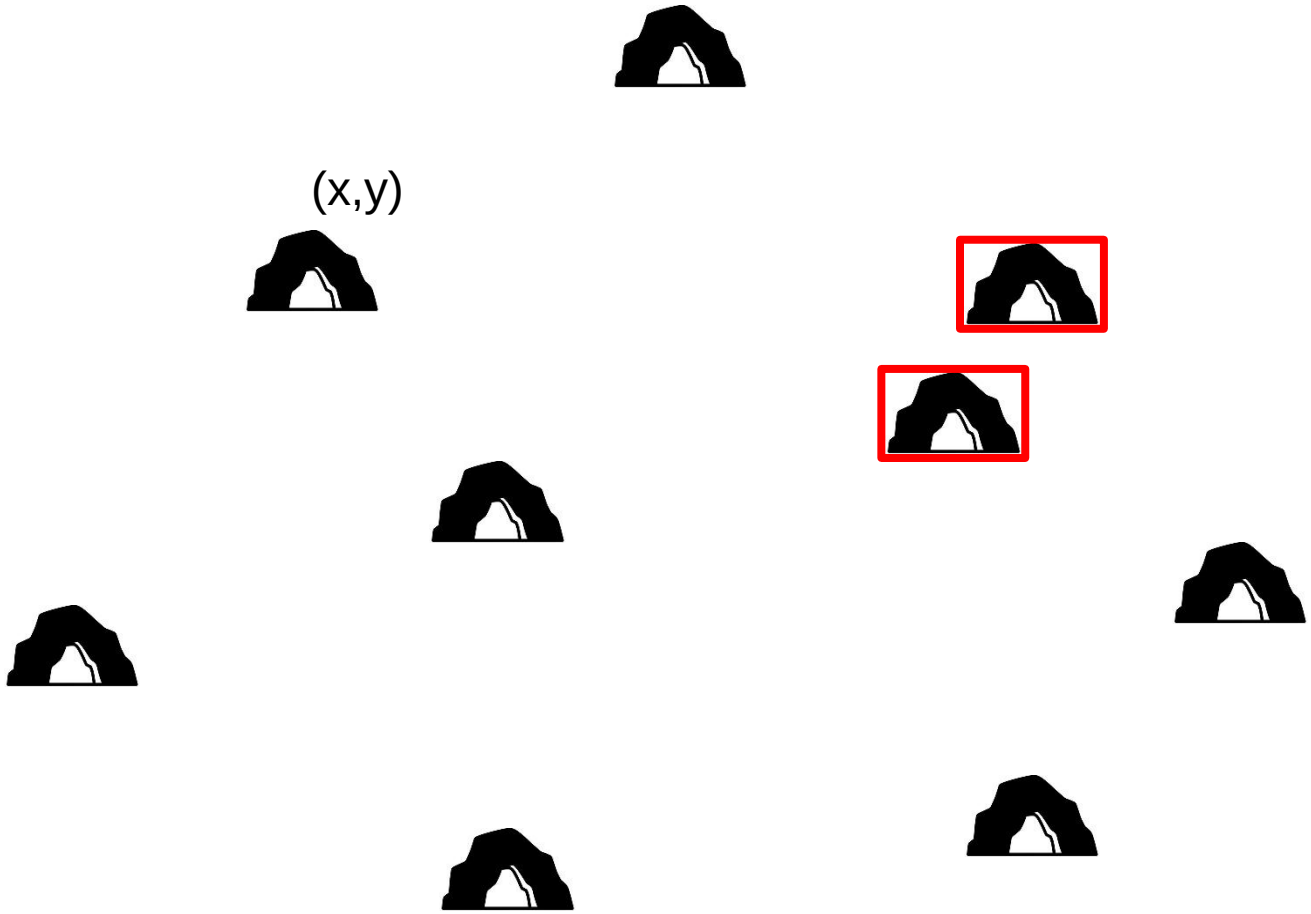


Given *n* cave entrances, where each cave entrance has an x,y coordinate, find the pair of caves with the smallest distance between them

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

Algorithm ?

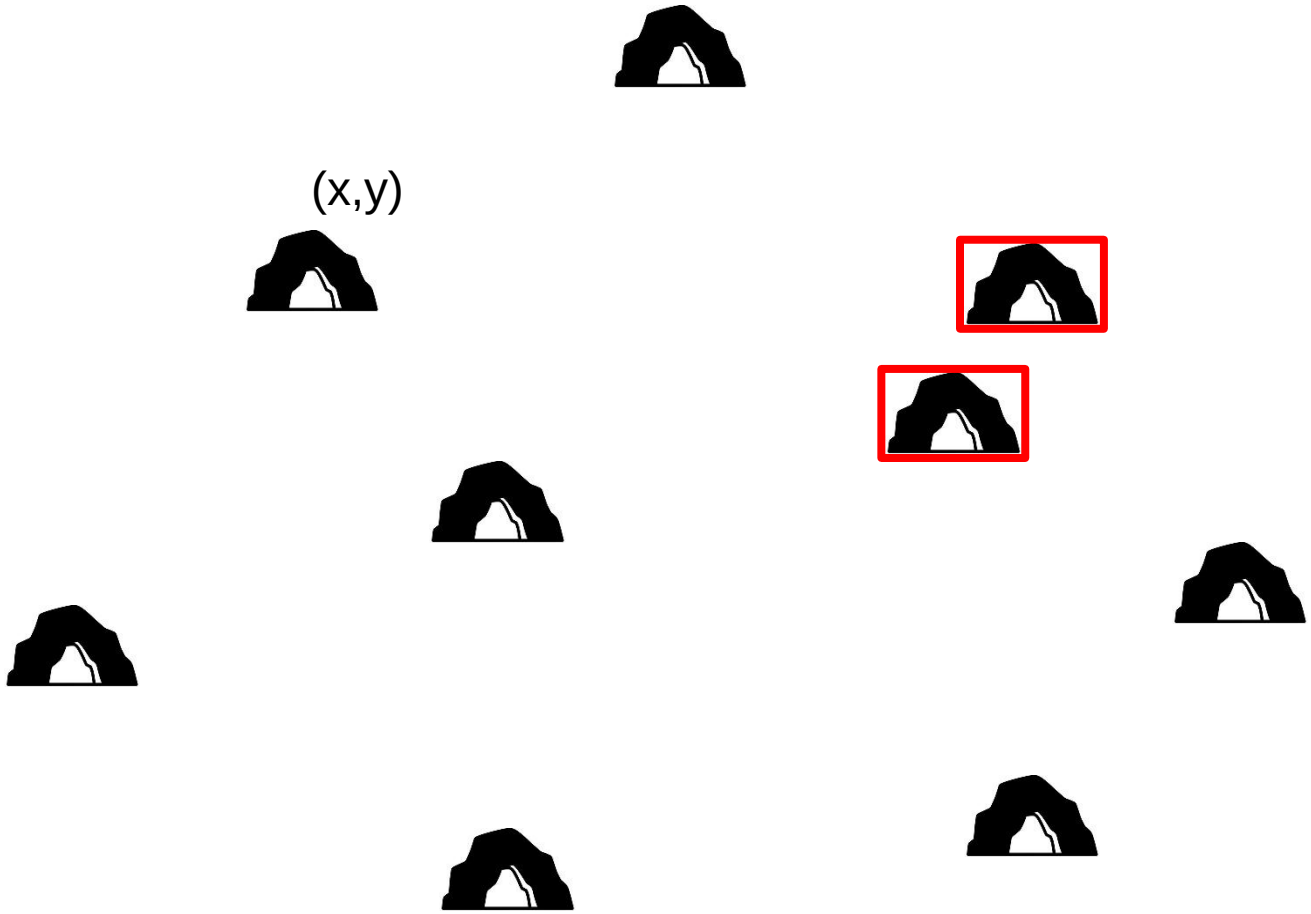
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



	C1	C2	C3	…	C9
C1	/	D(1,2)	D(1,3)	…	D(1,9)
C2	D(2,1)	/	D(2,3)	…	D(2,9)
C3	D(3,1)	D(3,2)	/	…	D(3,9)
…	…	…	…	…	….
C9	D(9,1)	D(9,2)	D(9,3)	…	/

Algorithm ?

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



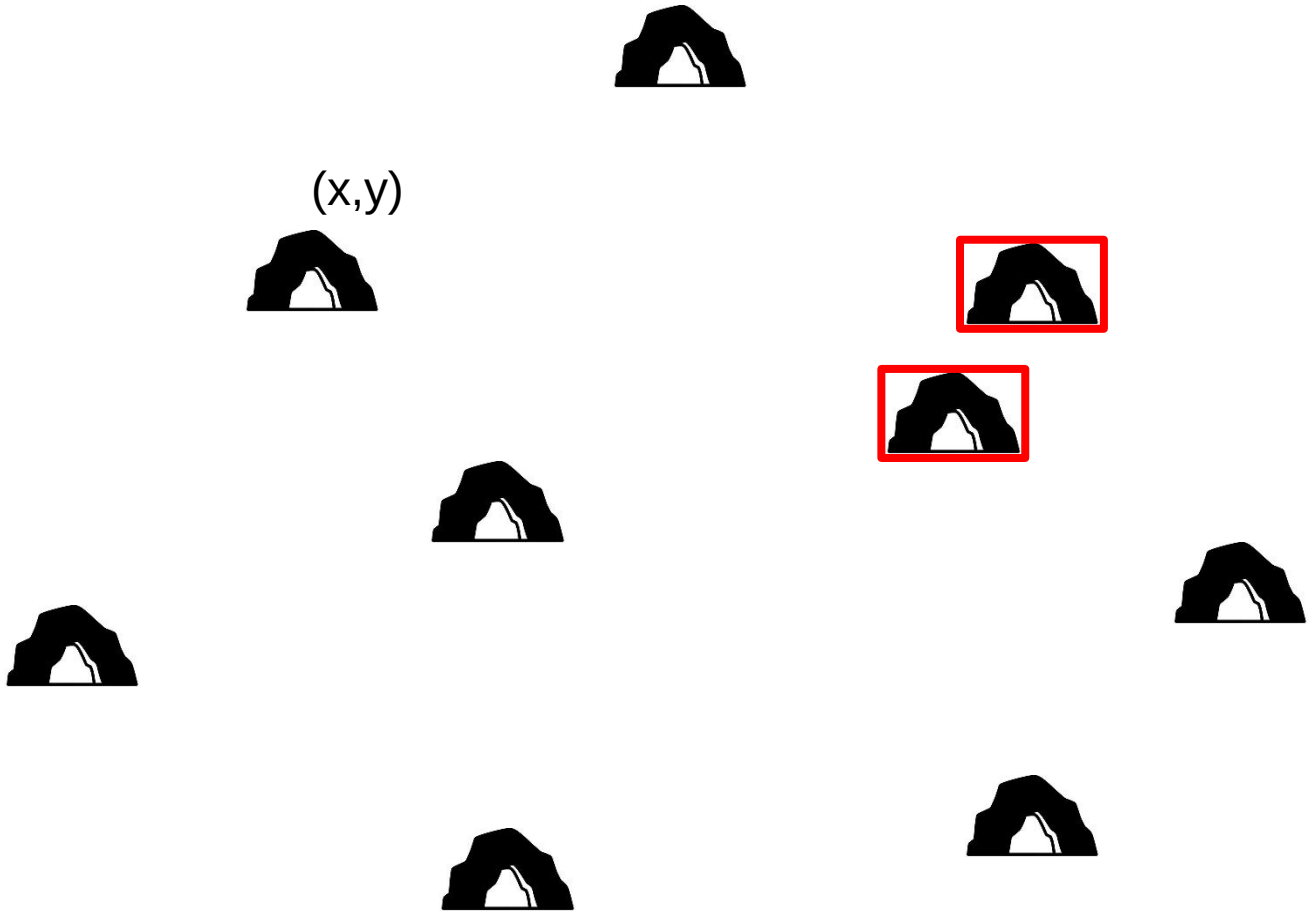
	C1	C2	C3	…	C9
C1	/	D(1,2)	D(1,3)	…	D(1,9)
C2	D(2,1)	/	D(2,3)	…	D(2,9)
C3	D(3,1)	D(3,2)	/	…	D(3,9)
…	…	…	…	…	….
C9	D(9,1)	D(9,2)	D(9,3)	…	/

Basic solution:

1. Compute distance for each pair
2. Select smallest

Algorithm ?

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

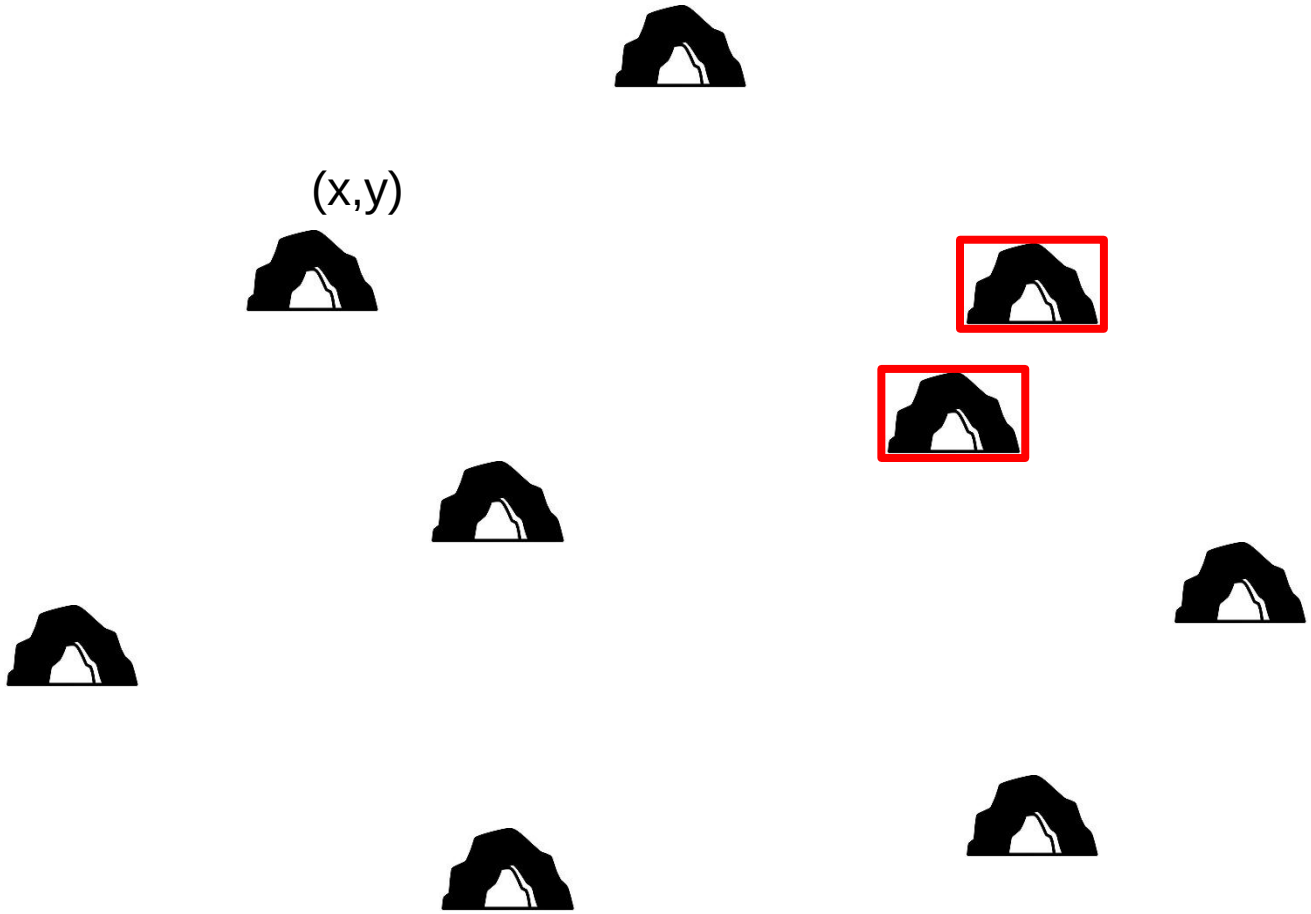


	C1	C2	C3	…	C9
C1	/	D(1,2)	D(1,3)	…	D(1,9)
C2	D(2,1)	/	D(2,3)	…	D(2,9)
C3	D(3,1)	D(3,2)	/	…	D(3,9)
…	…	…	…	…	….
C9	D(9,1)	D(9,2)	D(9,3)	…	/

Basic solution:

1. Compute distance for each pair
2. Select smallest

Running time = ?



	C1	C2	C3	…	C9
C1	/	D(1,2)	D(1,3)	…	D(1,9)
C2	D(2,1)	/	D(2,3)	…	D(2,9)
C3	D(3,1)	D(3,2)	/	…	D(3,9)
…	…	…	…	…	….
C9	D(9,1)	D(9,2)	D(9,3)	…	/

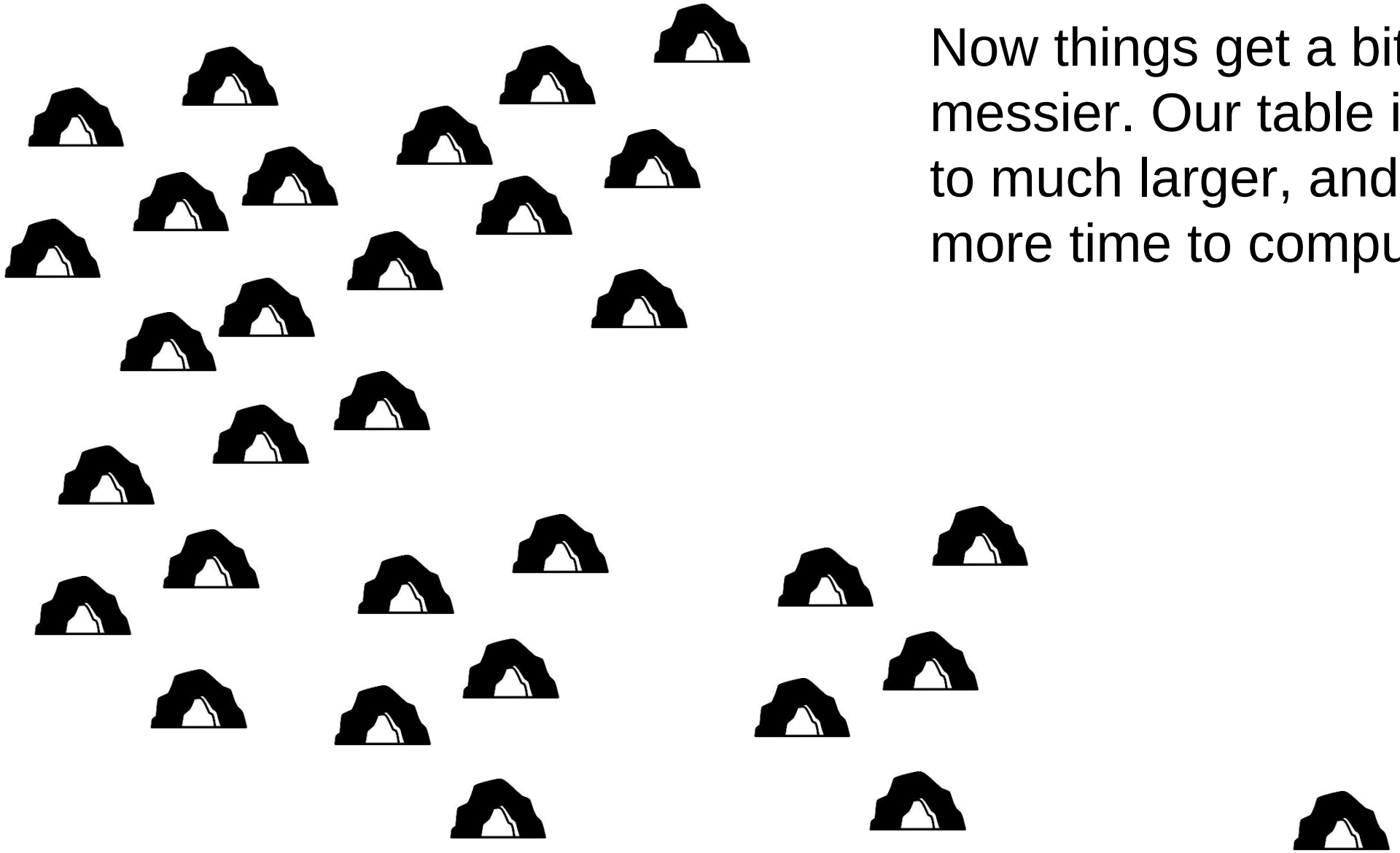
Basic solution:

1. Compute distance for each pair
2. Select smallest

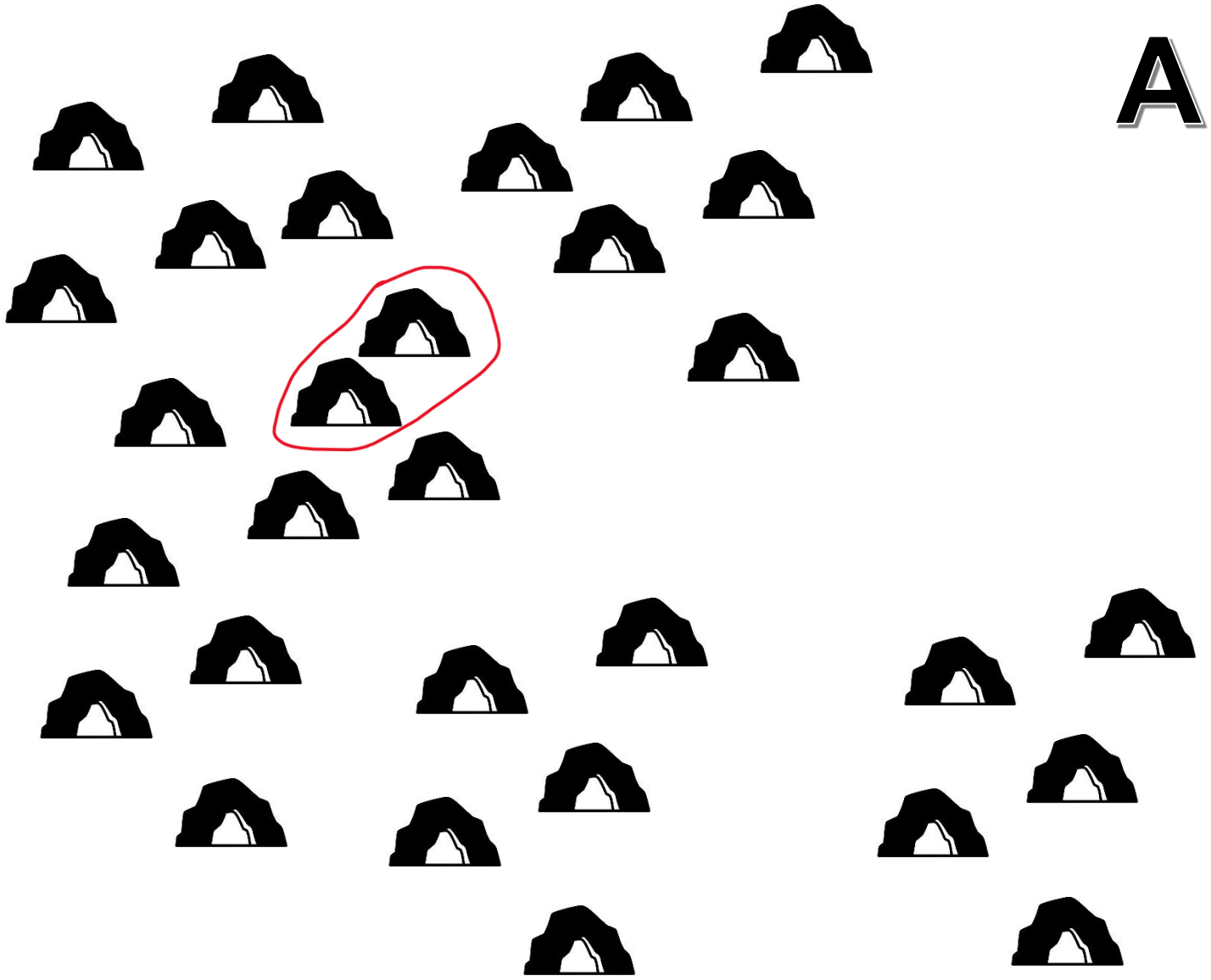
Running time = ?

Running time = $O(n^2)$

n = # of caves

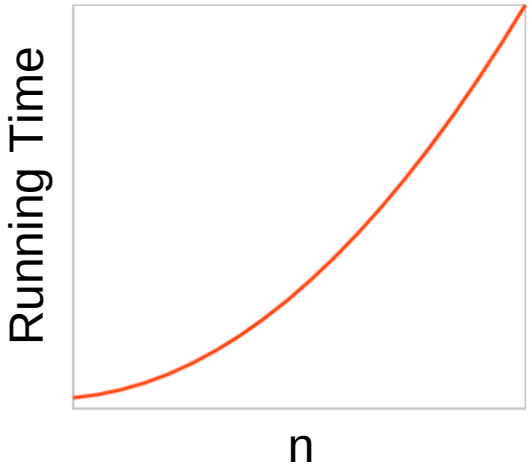


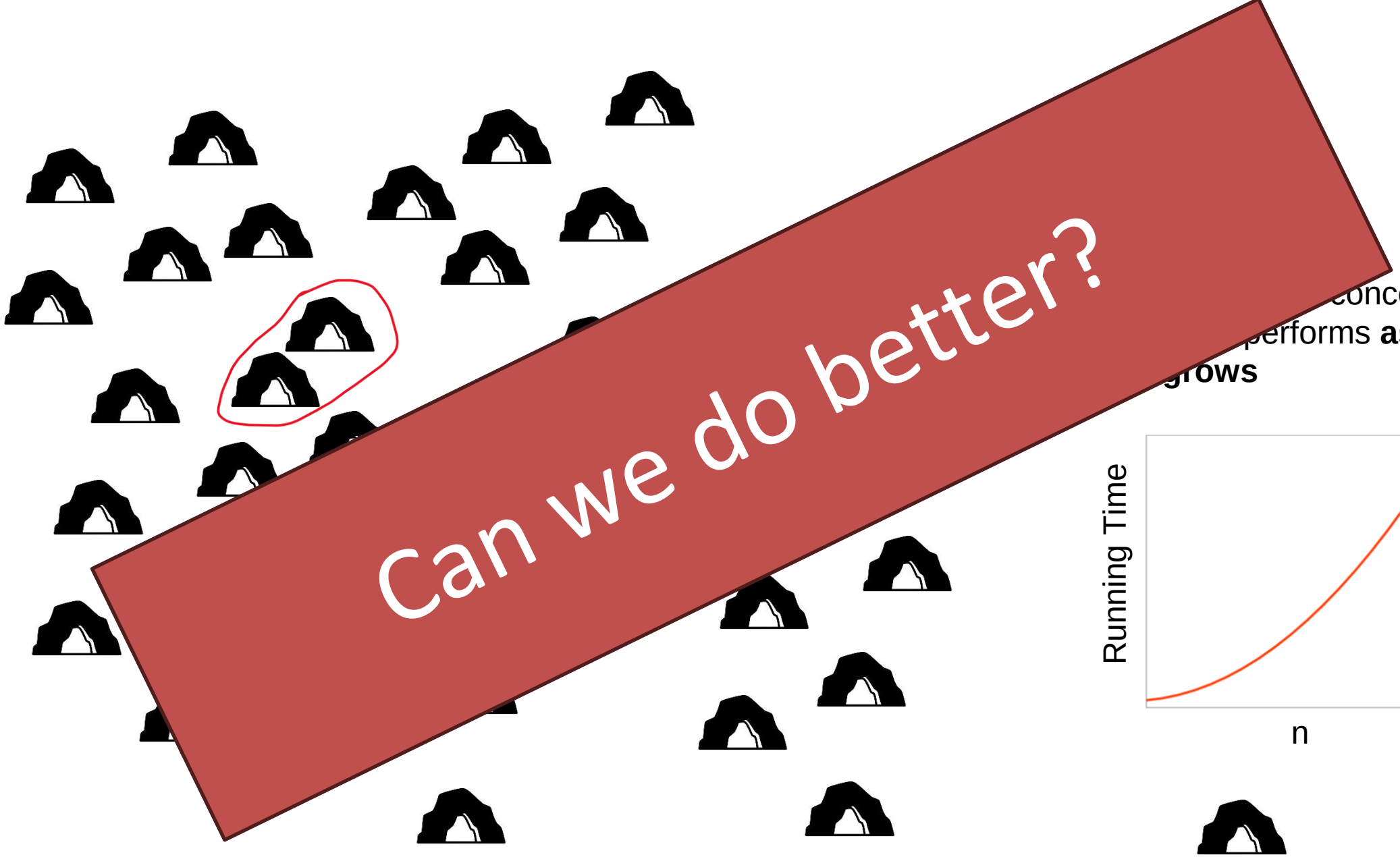
Now things get a bit messier. Our table is going to much larger, and will take more time to compute



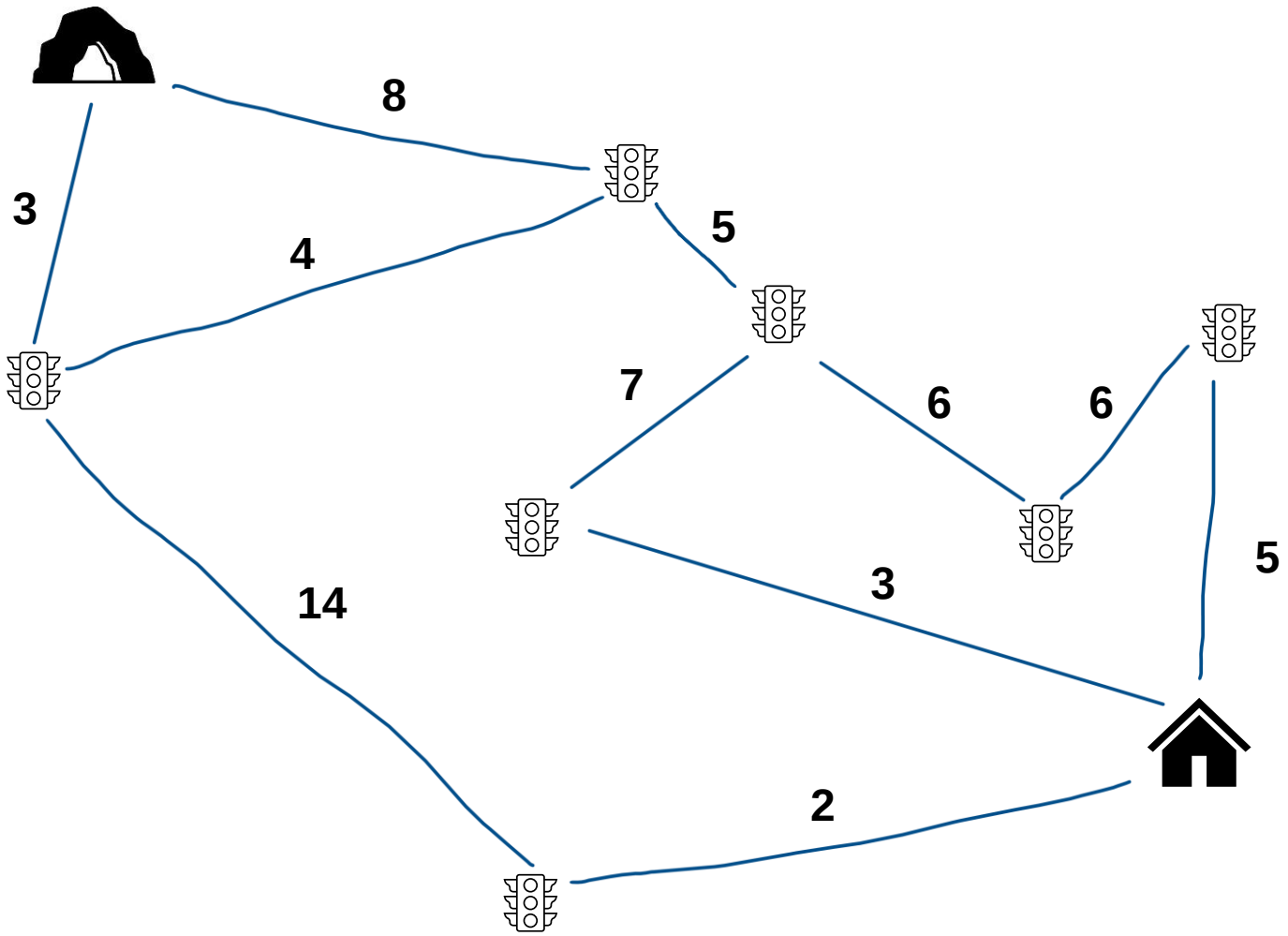
Algorithms

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



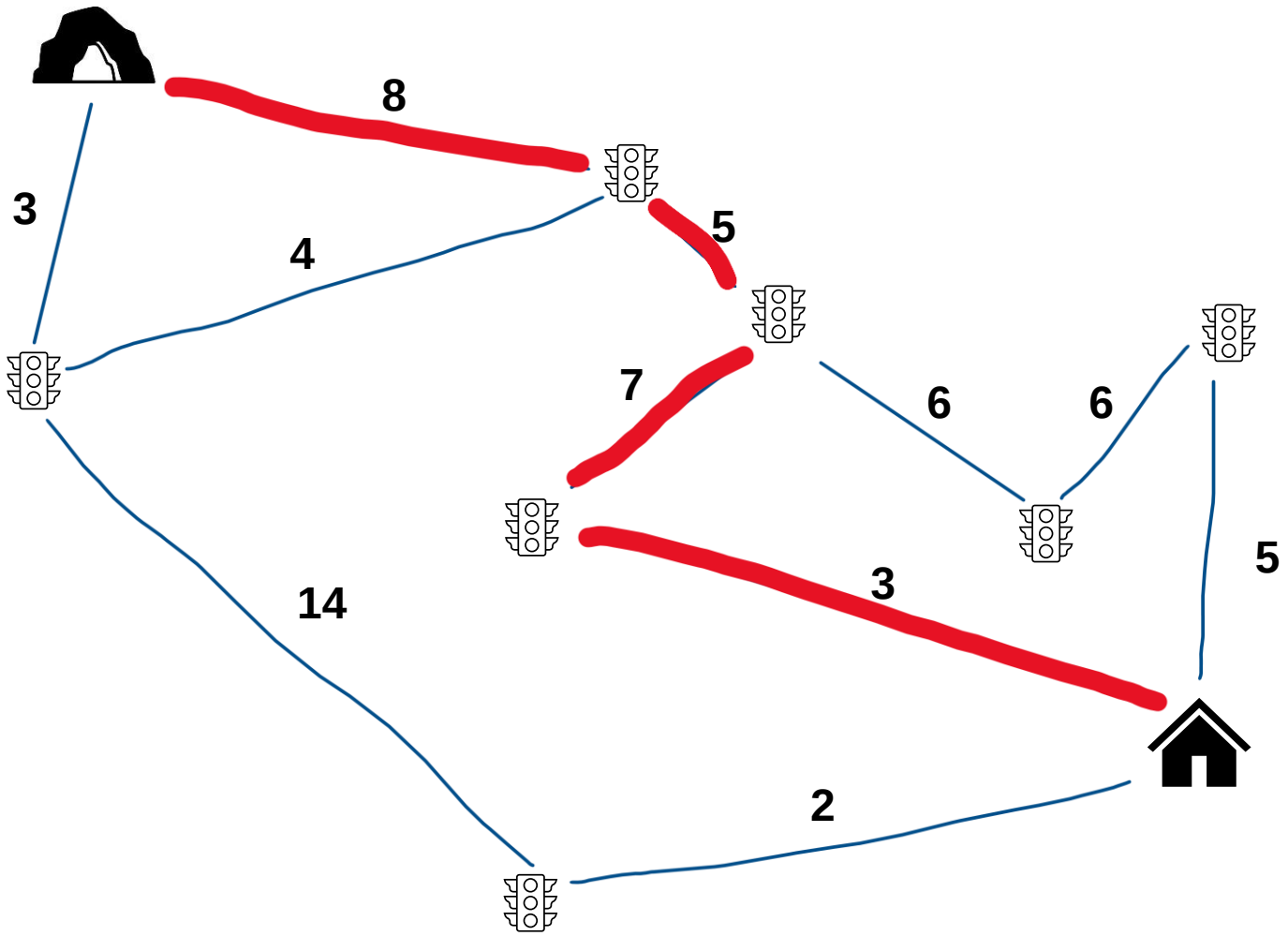


concerned with
performs **as some**
grows



We can reach the cave through a series of roads, each having a distance of *d*

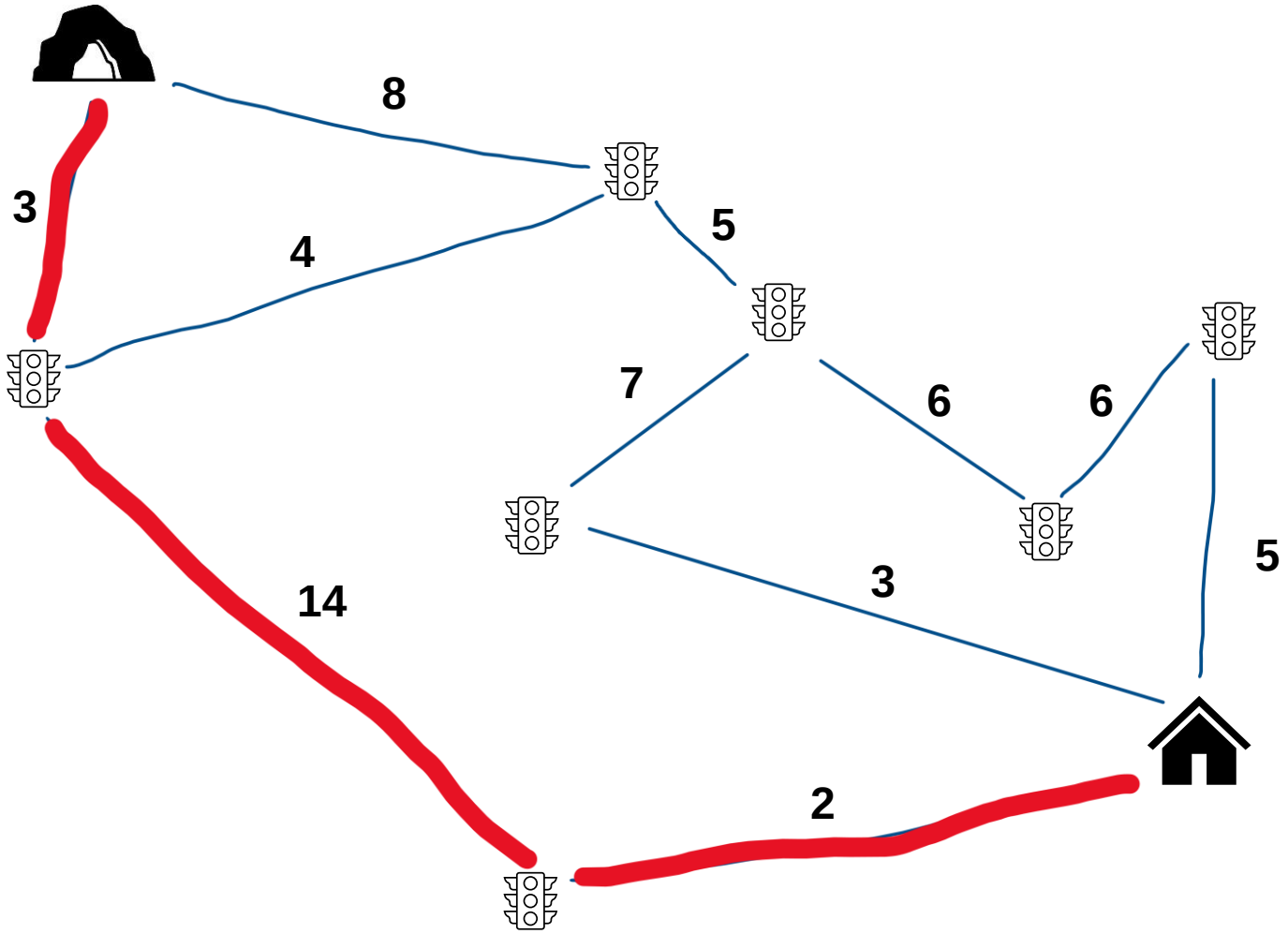
There are several ways to get there!



We can reach the cave through a series of roads, each having a distance of *d*

There are several ways to get there!

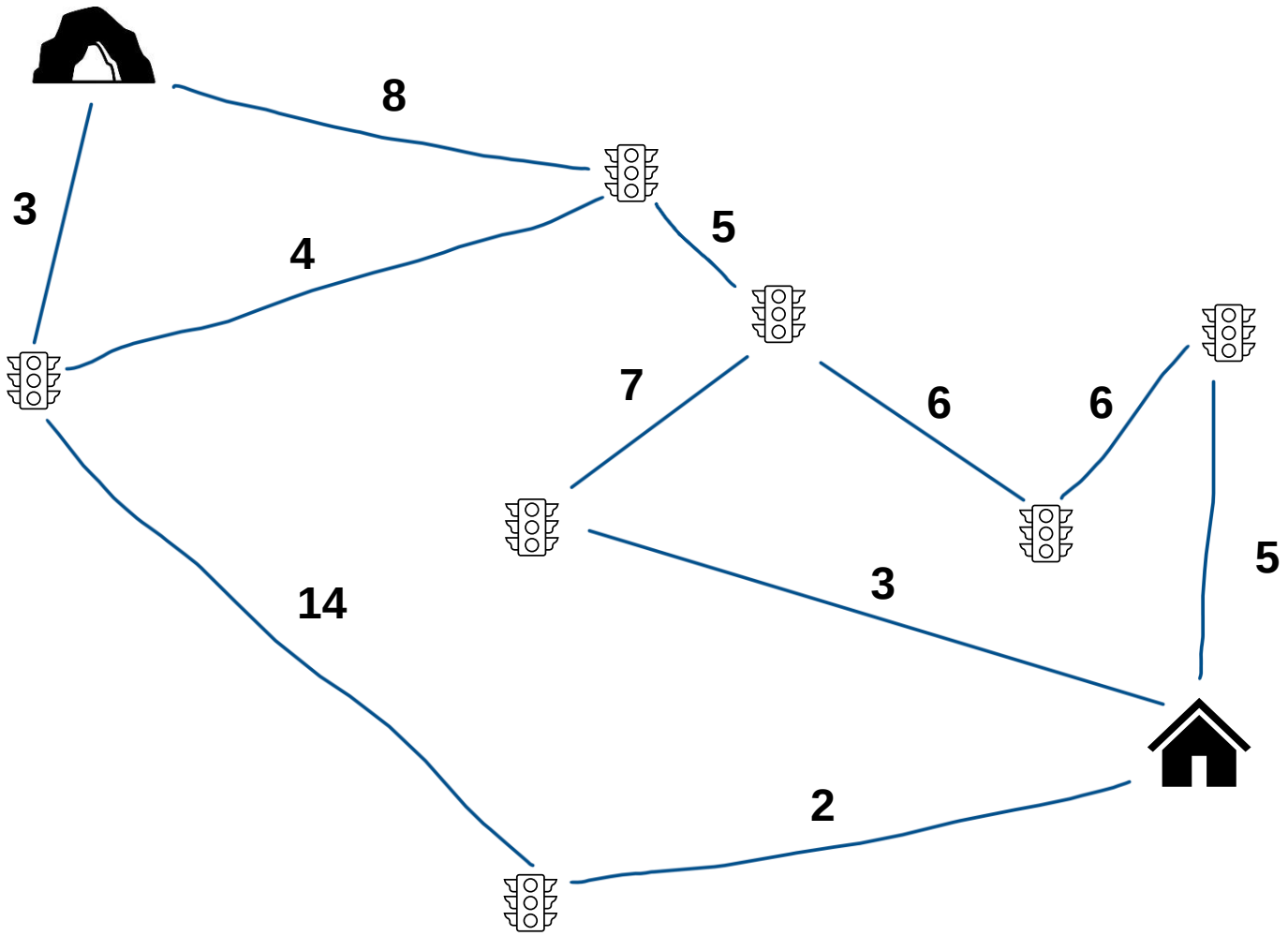
Cost: 23 (3 + 7 + 5 + 8)



We can reach the cave through a series of roads, each having a distance of *d*

There are several ways to get there!

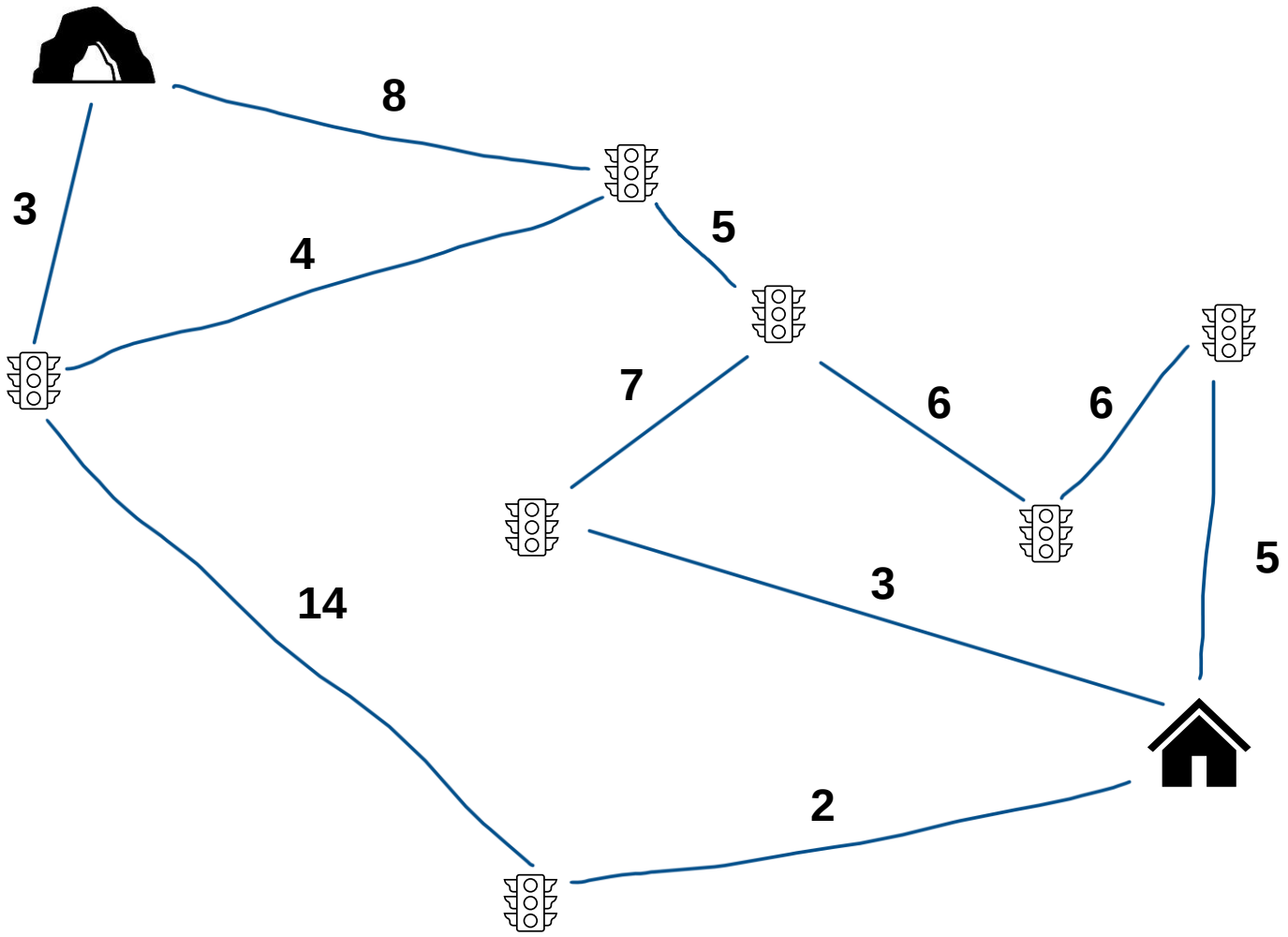
Cost: 19 (2 + 14 + 3)



We can reach the cave through a series of roads, each having a distance of *d*

There are several ways to get there!

What is the shortest path from start to finish?



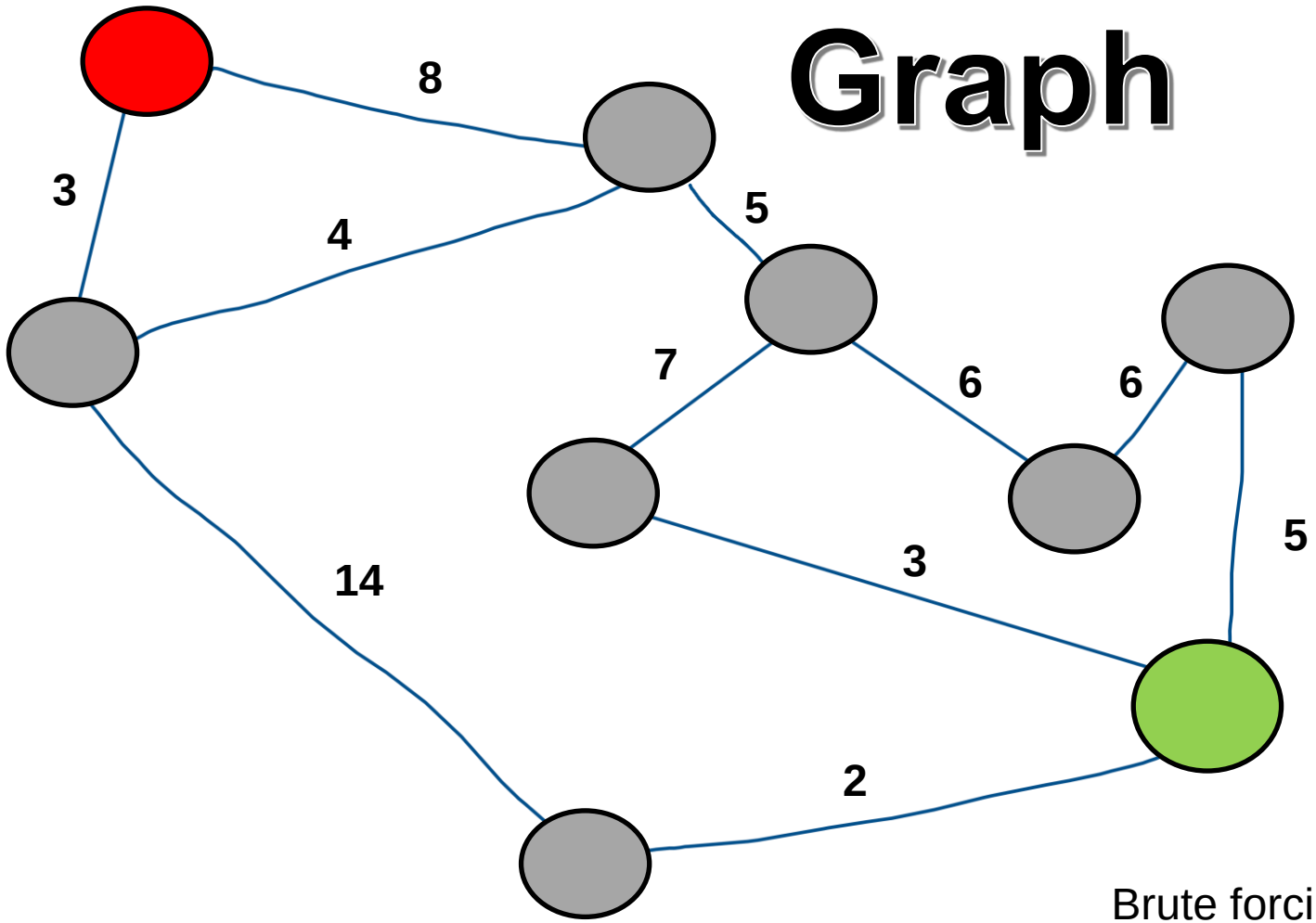
We can reach the cave through a series of roads, each having a distance of *d*

There are several ways to get there!

What is the shortest path from start to finish?

Brute Force?

Graph



We can reach the cave through a series of roads, each having a distance of *d*

There are several ways to get there!

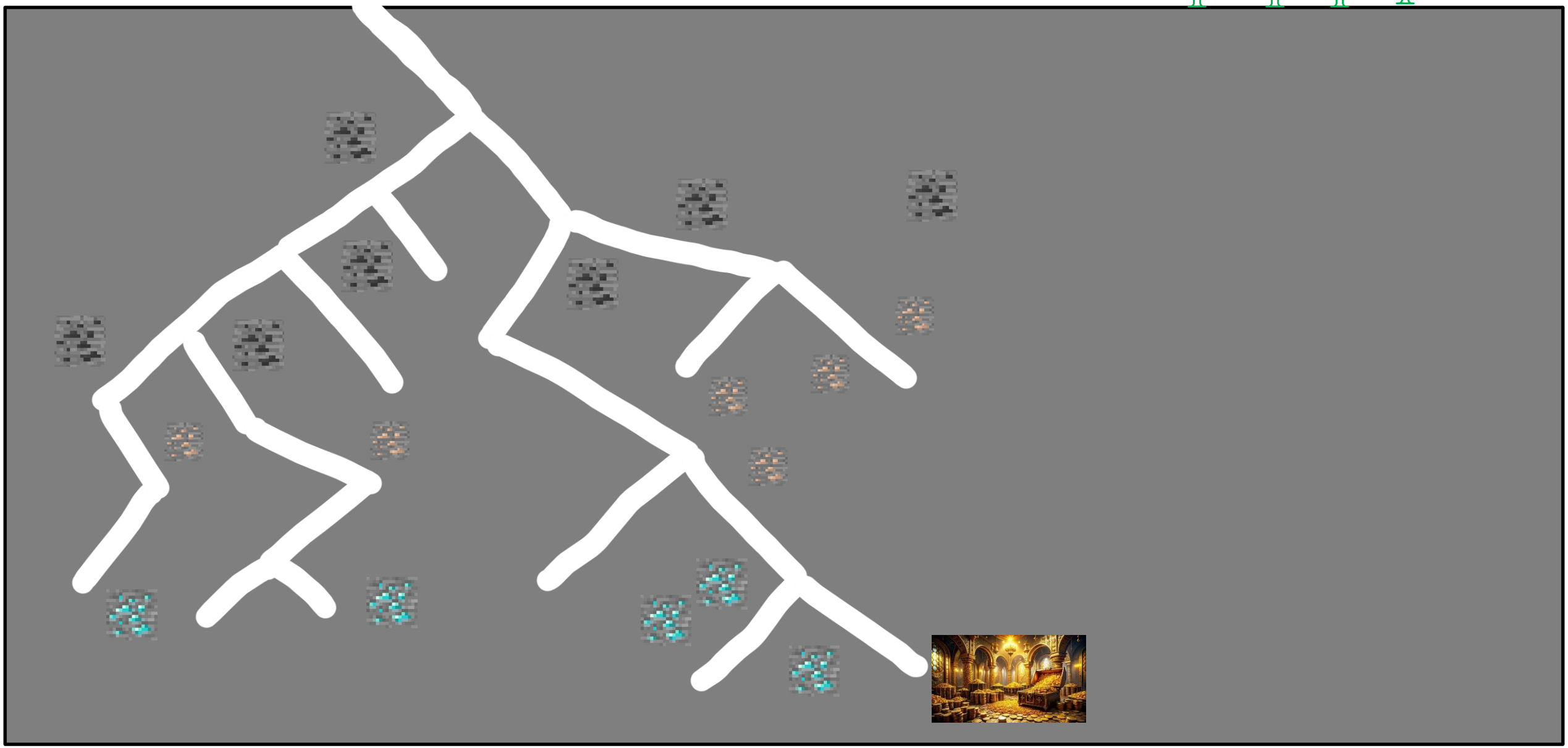
What is the shortest path from start to finish?

Brute Force?

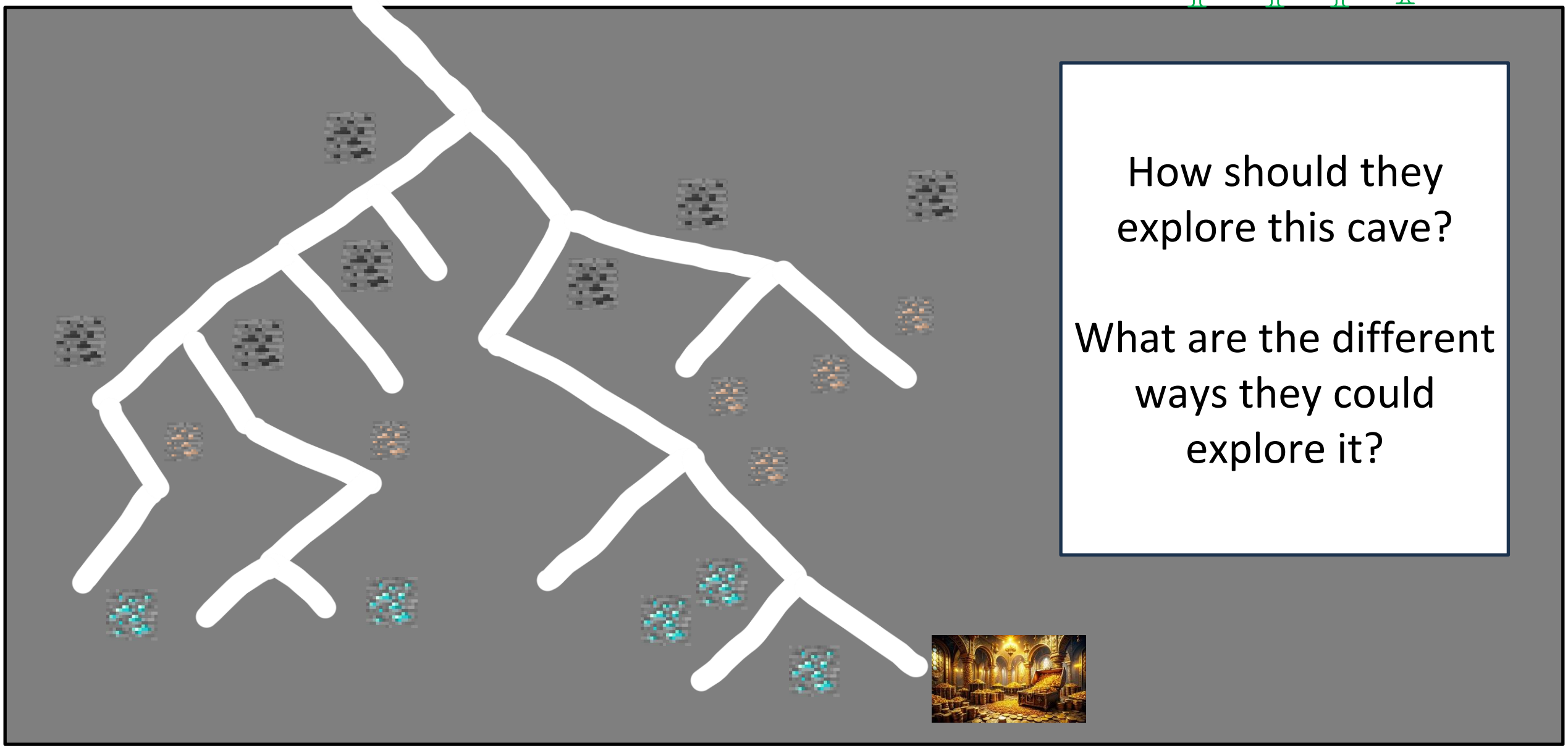
Brute forcing every possible path is **not feasible**
(ie exponential or factorial time)

It is ok for this map, but we want our approach to work on every possible map

We have now reached the cave, but the cave consists of many complex tunnels



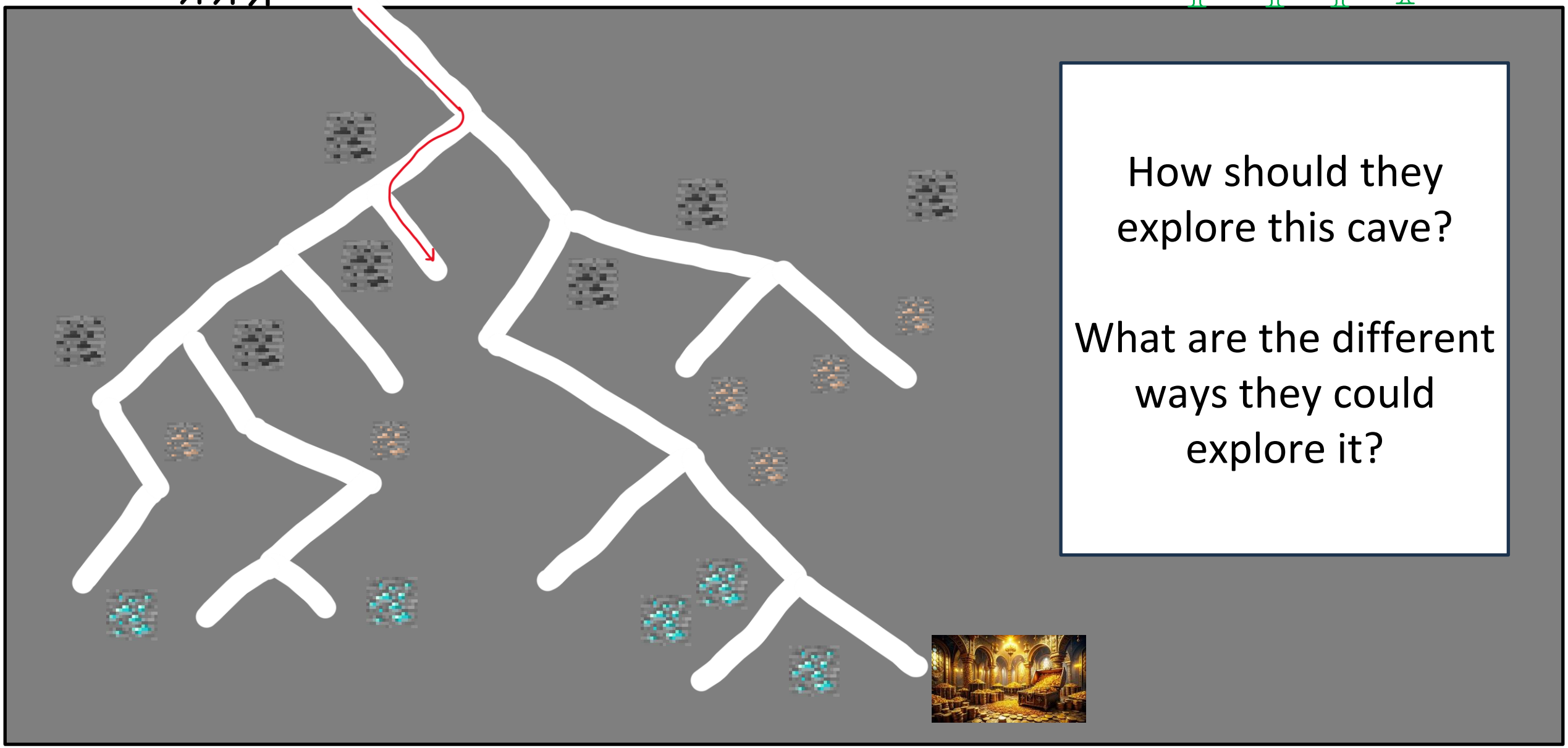
We have now reached the cave, but the cave consists of many complex tunnels



How should they explore this cave?

What are the different ways they could explore it?

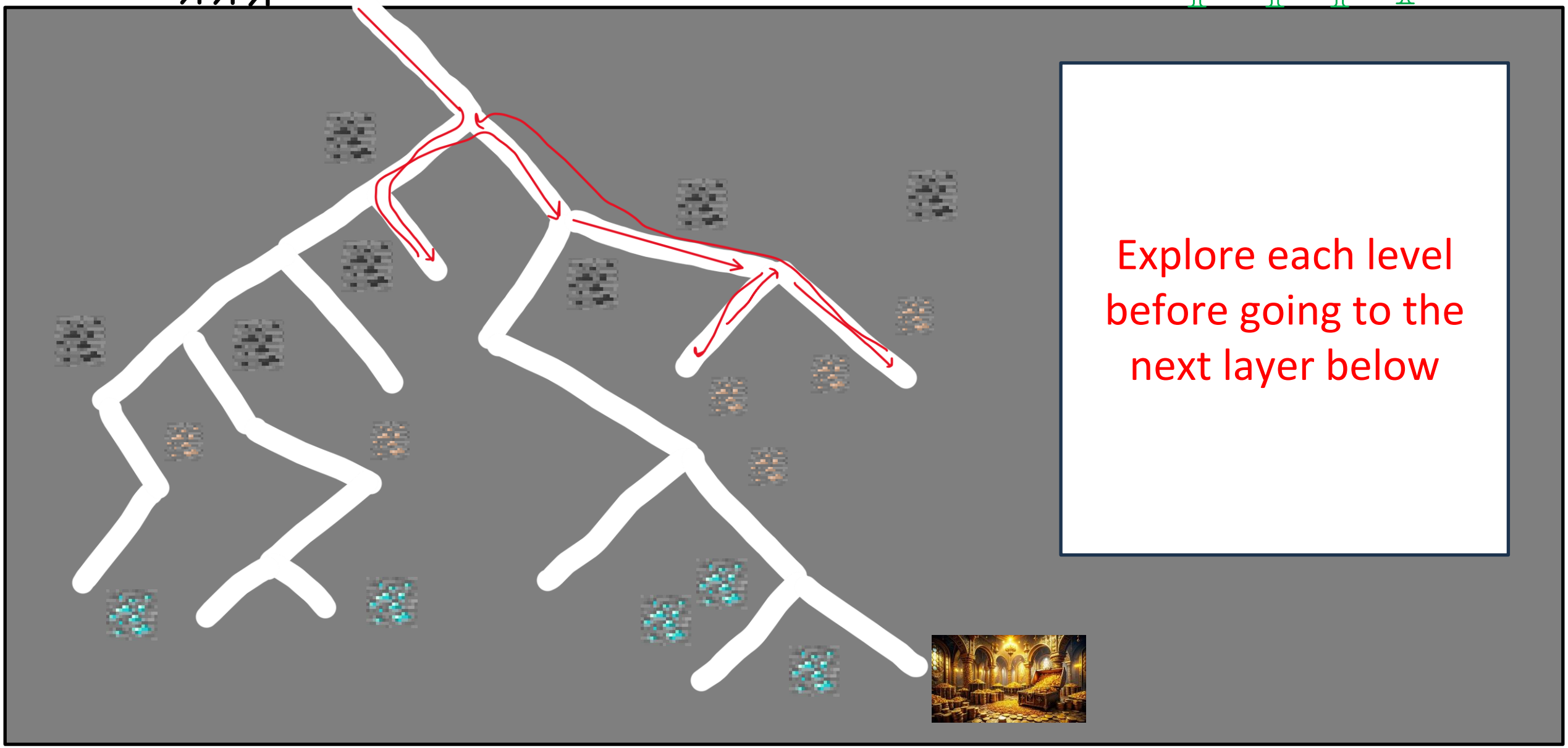
We have now reached the cave, but the cave consists of many complex tunnels



How should they explore this cave?

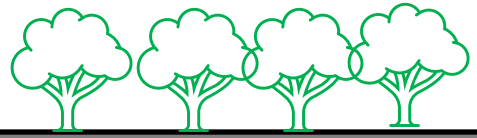
What are the different ways they could explore it?

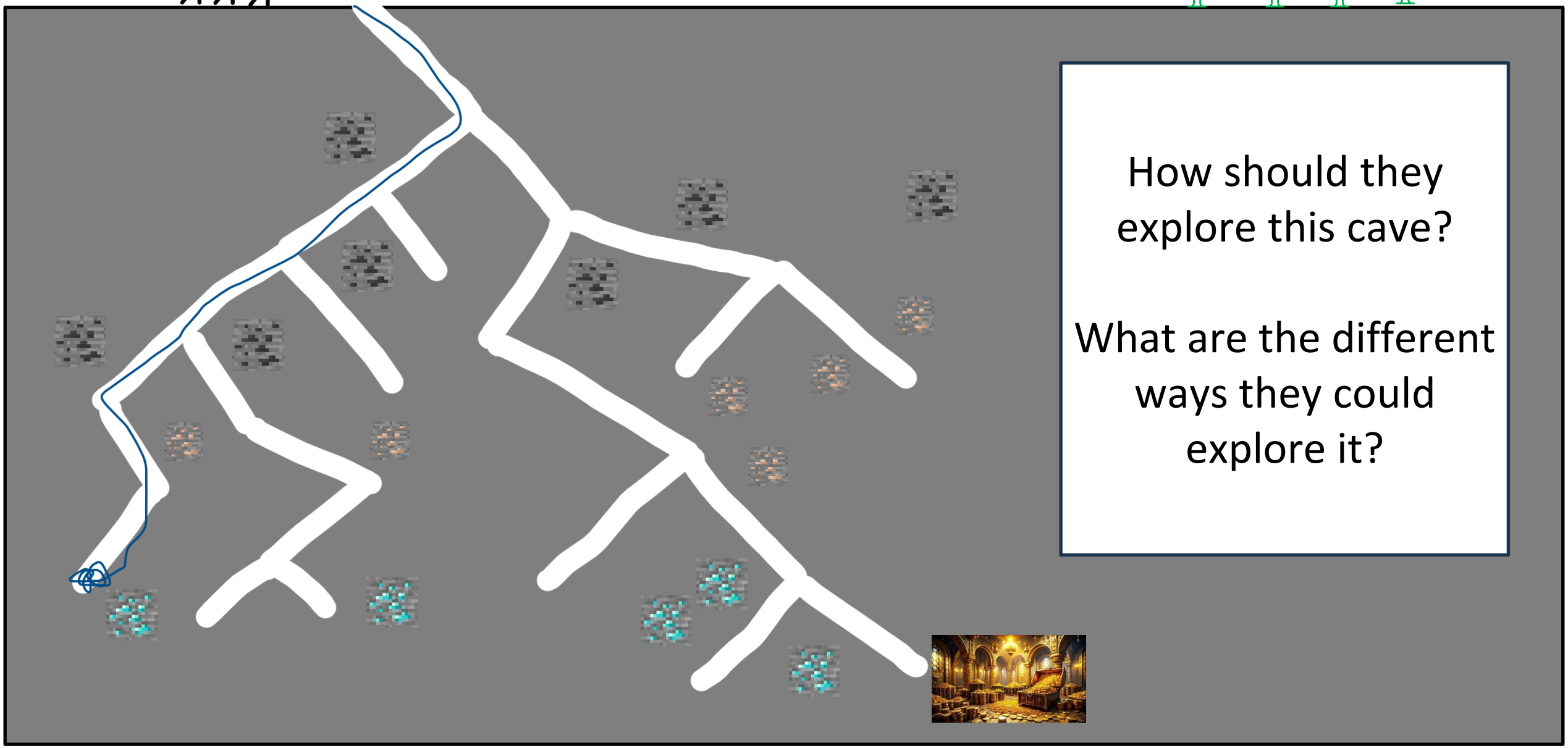
We have now reached the cave, but the cave consists of many complex tunnels



Explore each level before going to the next layer below

We have now reached the cave, but the cave consists of many complex tunnels

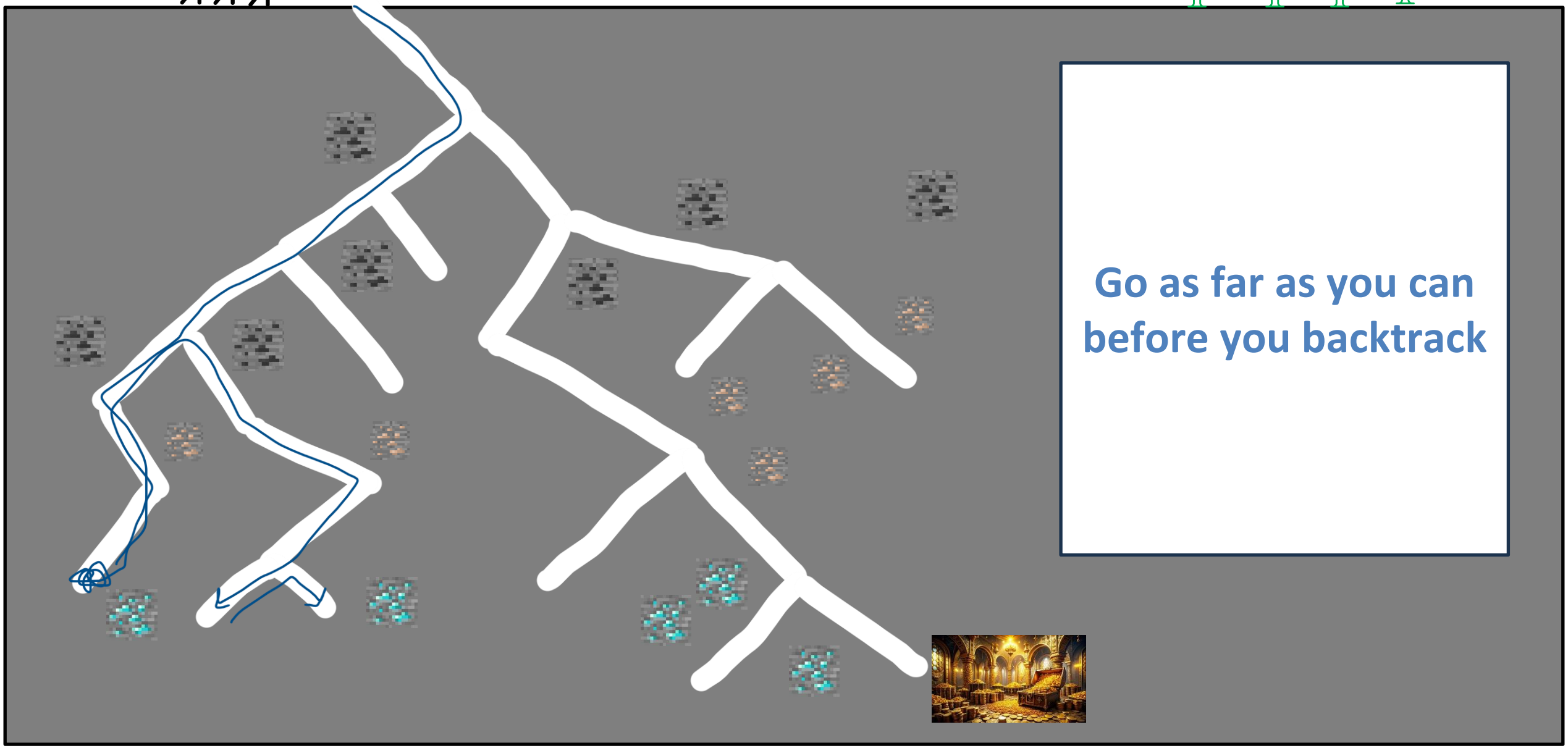




How should they explore this cave?

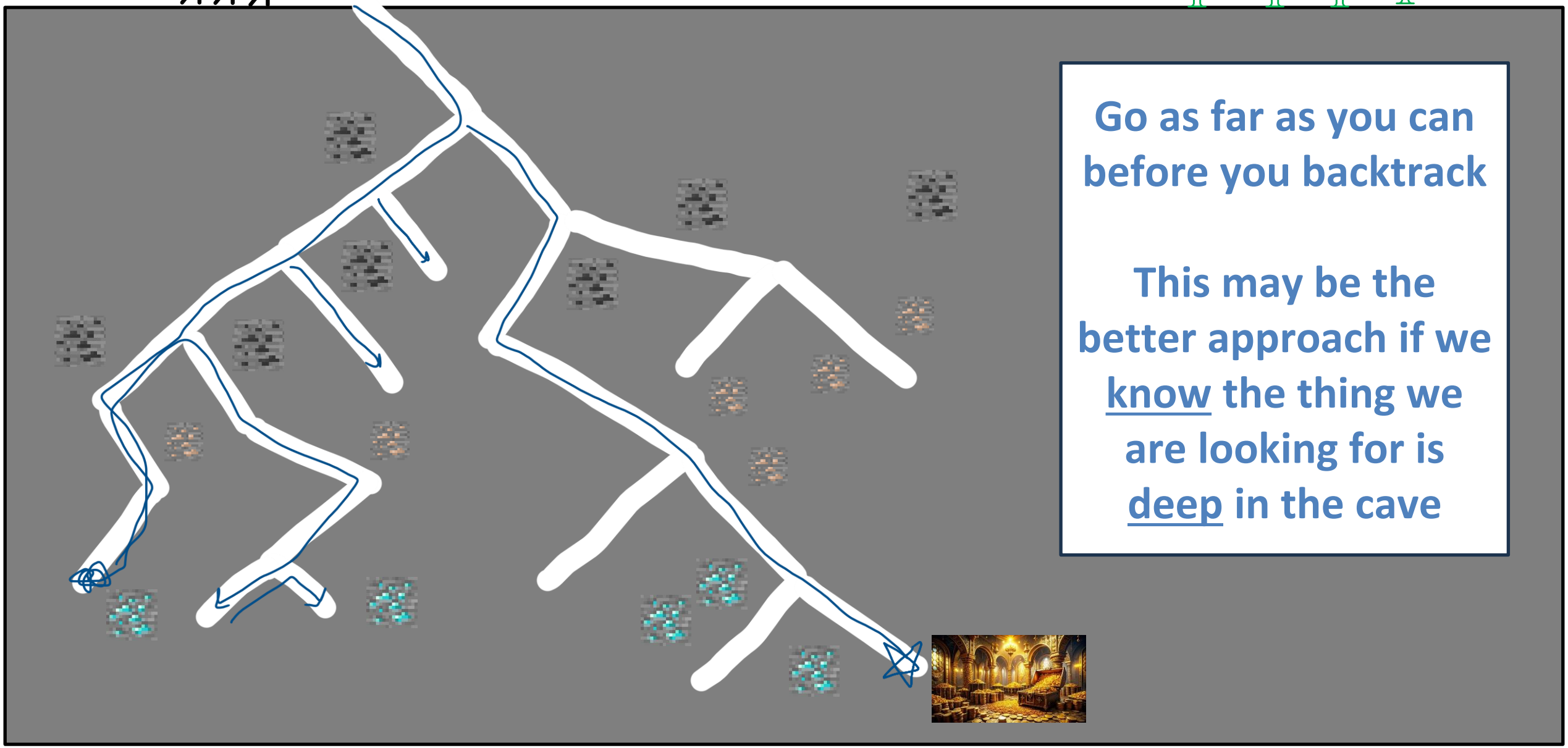
What are the different ways they could explore it?

We have now reached the cave, but the cave consists of many complex tunnels



Go as far as you can before you backtrack

We have now reached the cave, but the cave consists of many complex tunnels



Go as far as you can before you backtrack

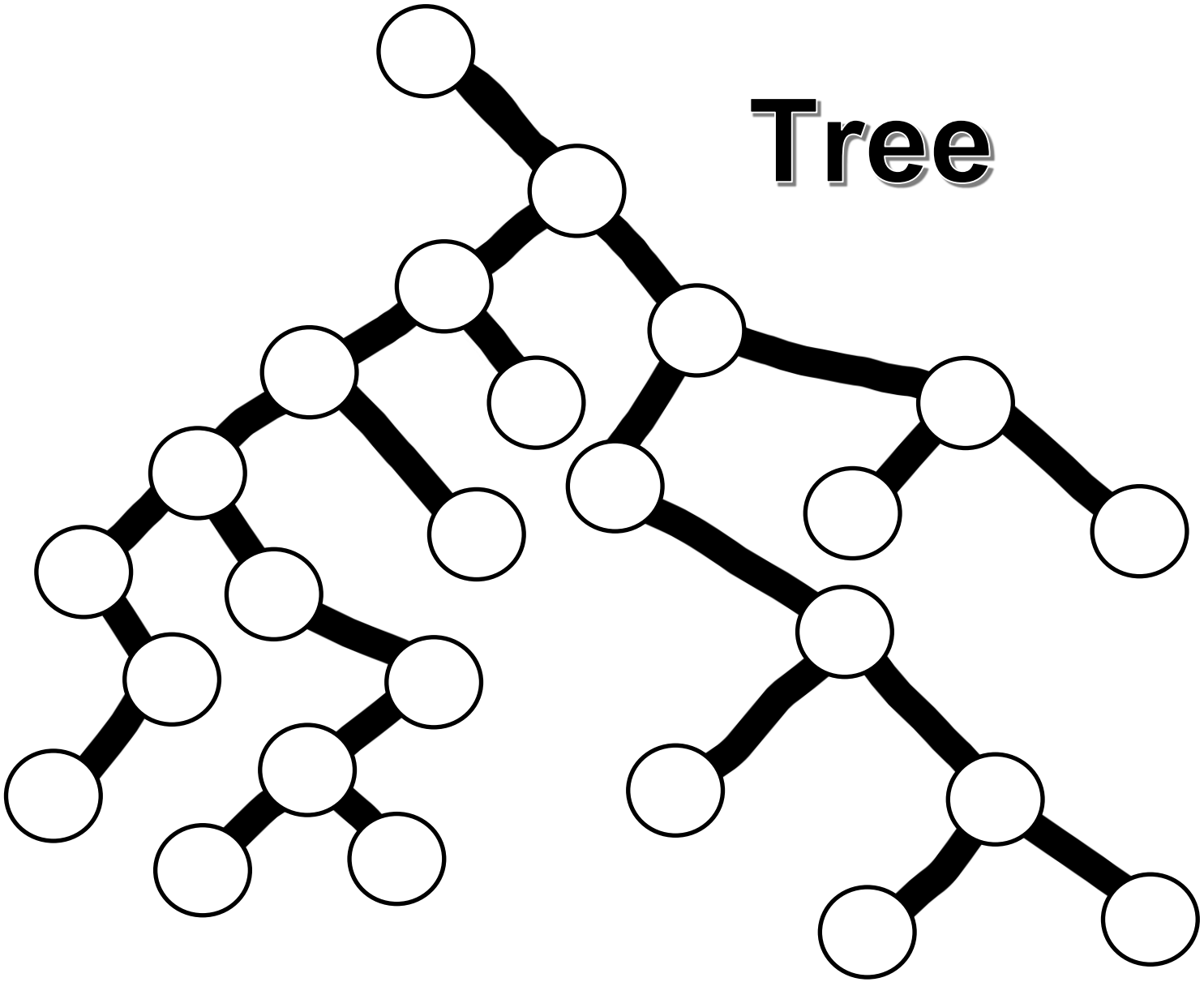
This may be the better approach if we know the thing we are looking for is deep in the cave

We have now reached the cave, but the cave consists of many complex tunnels

Explore each level before going to the next layer below

Go as far as you can before you backtrack

We have now reached the cave, but the cave consists of many complex tunnels



Tree Traversal Methods

Explore each level before going to the next layer below

Go as far as you can before you backtrack

We have reached the treasure room, but there is much more treasure than our backpack can fit

We must pick what treasure we want to take !

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

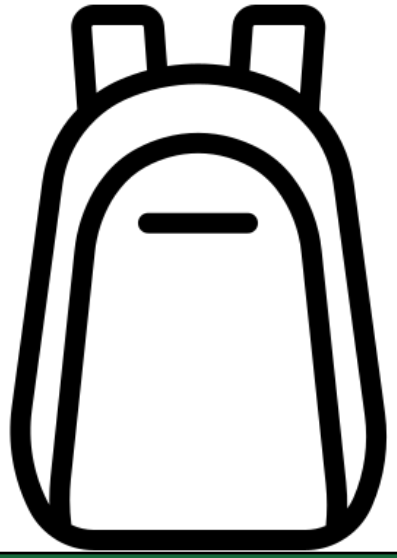
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items fand 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

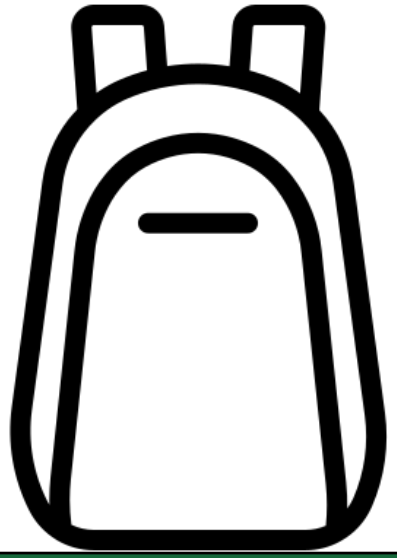
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items fand 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

Total value: 70
Weight: 10 pounds ✓

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

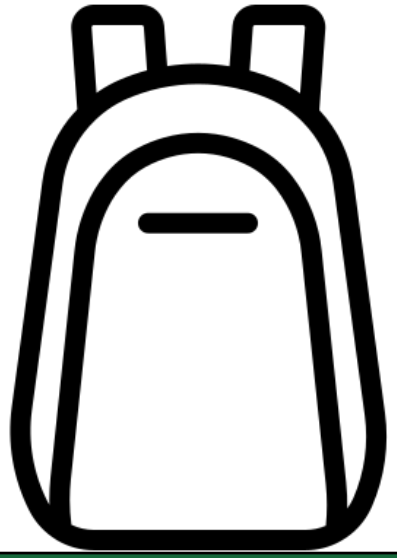
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items fand 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

Algorithm?

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

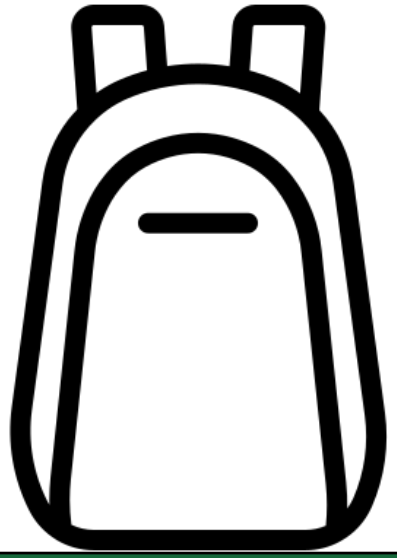
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items fand 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

Algorithm?

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

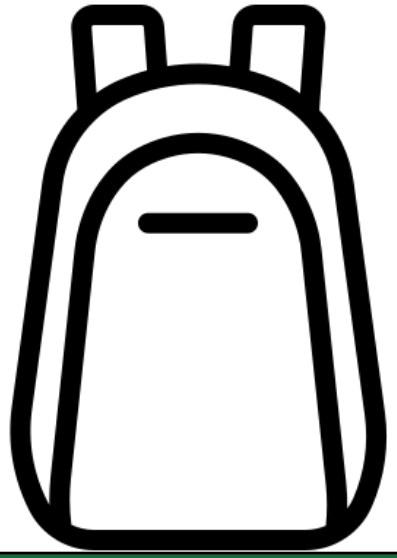
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items fand 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

Algorithm?

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

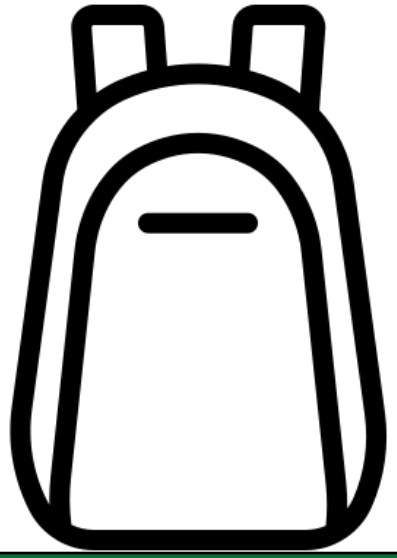
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items fand 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: 95
Weight: 9 ✓

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

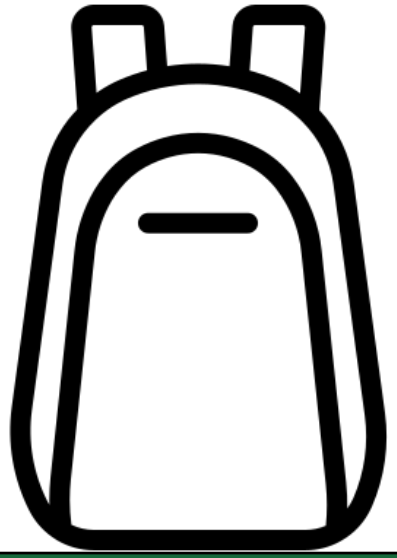
Value: 50
Weight: 3

Value: 5
Weight: 2

We are going to steal some items fand 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

Is this the **optimal** solution?

Value: 95
Weight: 9 ✓

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

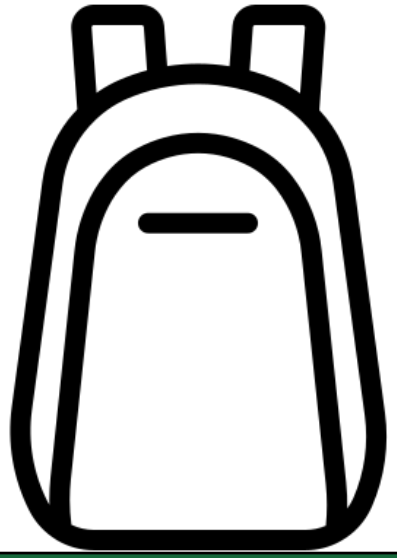
Value: 50
Weight: 8

Value: 5
Weight: 2

We are going to steal some items fand 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

Is this the **optimal** solution?

Value:
Weight:

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

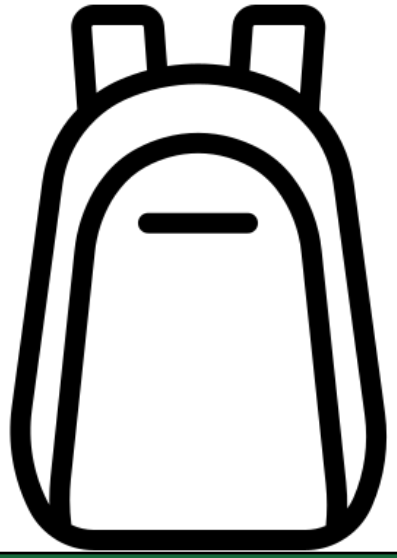
Value: 50
Weight: 8

Value: 5
Weight: 2

We are going to steal some items fand 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

Is this the **optimal** solution?

Value: 55
Weight: 10

Value: 10
Weight: 5

Value: 40
Weight: 4

70

Value: 30
Weight: 6

Value: 25
Weight: 4

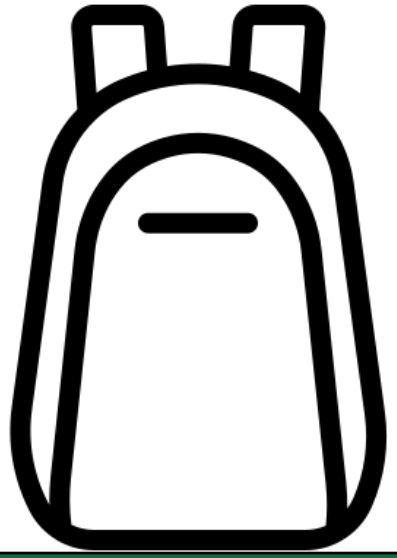
Value: 50
Weight: 8

Value: 5
Weight: 2

We are going to steal some items fand 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

Is this the **optimal** solution?

Value: 55
Weight: 10

X

Value: 10
Weight: 5

Value: 40
Weight: 4

Value: 30
Weight: 6

Value: 25
Weight: 4

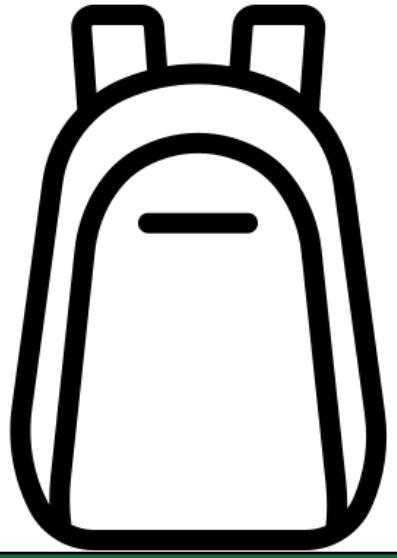
Value: 50
Weight: 8

Value: 5
Weight: 2

We are going to steal some items fand 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

Optimization

Suppose we want to keep a record of which person has which treasures, ie a **ledger**

Reese: 

Bob: 

Iliana: 

Suppose we want to keep a record of which person has which treasures, ie a **ledger**

Reese: 

Bob: 

Iliana: 

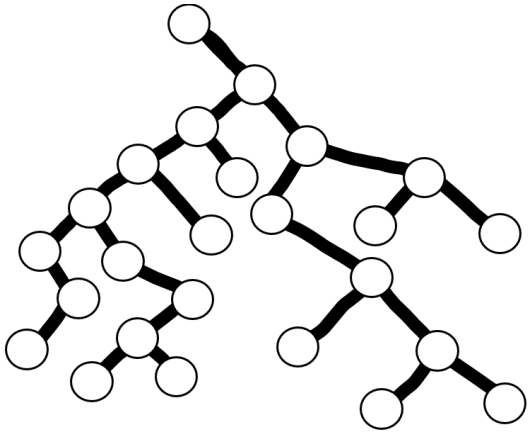
Key-Value Pairs

Name : Bag

Hash Tables

CSCI 232

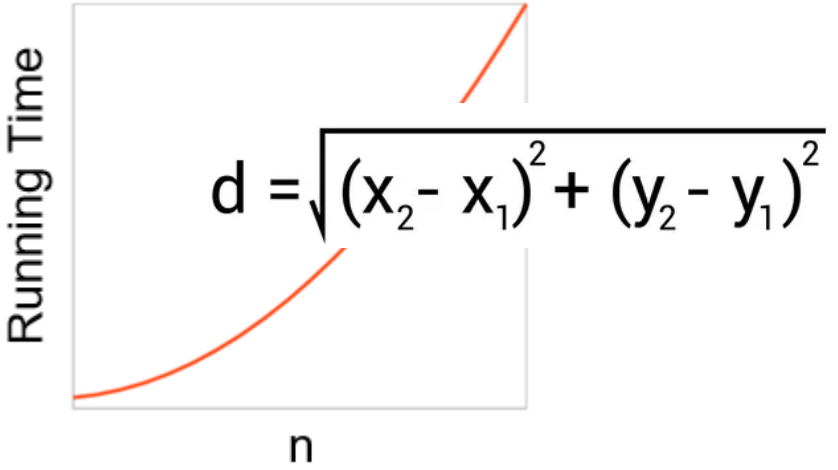
Trees



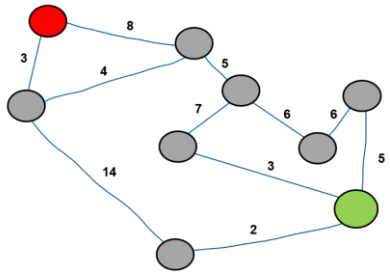
Optimization

Value: 95
Weight: 9 ✓

Algorithms



Graphs



Hash Tables

Reese: 

Bob: 

Iliana: 

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
		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

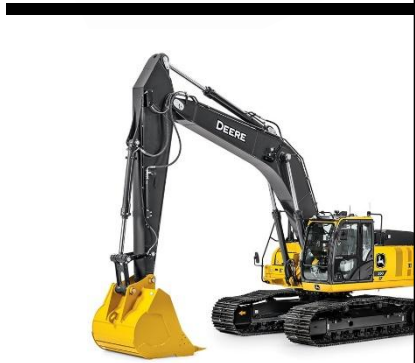
Best tool for the job?

a Well for

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

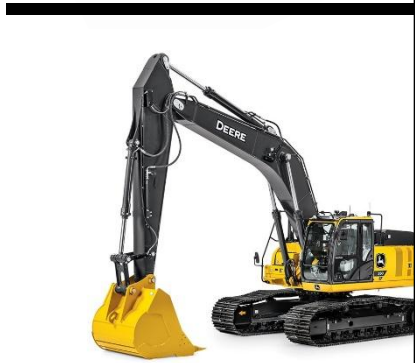
…es up a lot of
…ge space

What do you need to dig a hole?



of

What do you need to dig a hole?



We can't use the best tool for the job unless we know <u>how to use</u> that tool

CSCI 232- Data Structures and Algorithms

“Tools”

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

“Use of tools”

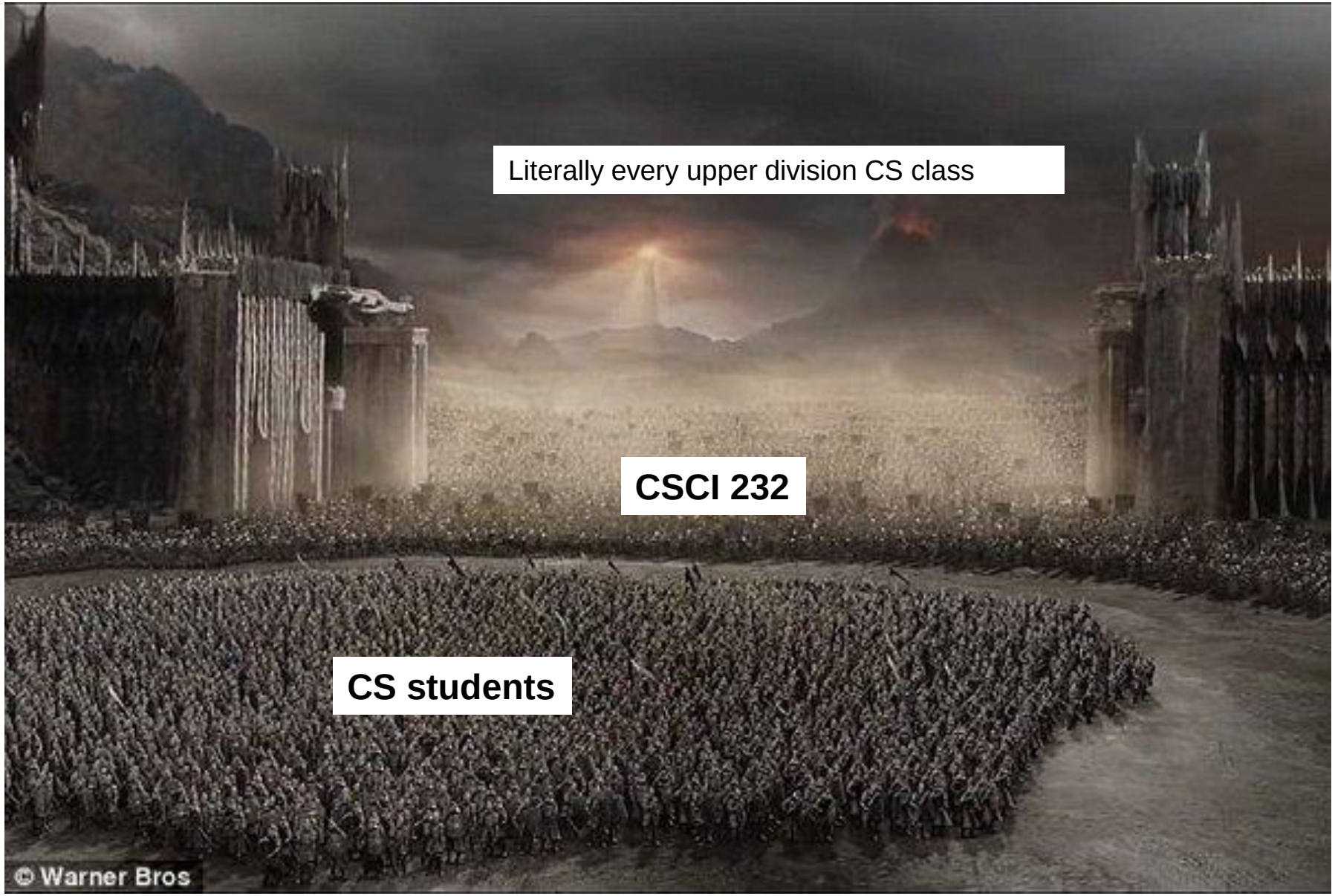
- Sorting
- Searching
- Routing
- Optimization

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
- I *want* this class to challenge you !


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

Reese Pearsall (pierce-all)

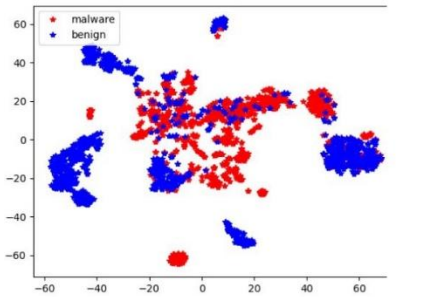
You can just call me "Reese" ☺

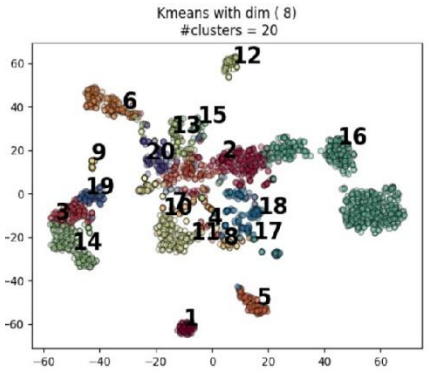
Fourth year Instructor @MSU
B.S & M.S @ MSU

Hometown
- Billings, MT

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

Teaching
- CSCI 232
- CSCI 476





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 (rip), Dr Pepper, Memes, *The Bachelor*, Naps

Contact

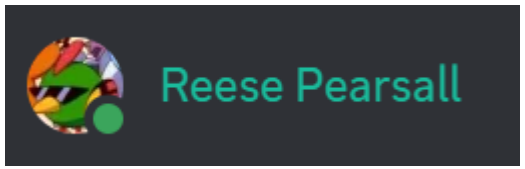
Email: reese.pearsall@montana.edu (I will respond as soon as I can)

Office Hours: Tuesday, Wednesday, Thursday 3:30 PM – 4:30 PM
and by appointment

You can send me an email/Discord DM for a virtual meeting room

Office: Barnard Hall 361 or Barnard Hall 254

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



Grader

Sarker Safat Mahmud (Safat)
Email: **sarkersafatmahmud@montana.edu**

He will be grading your programming assignments

Course Logistics (Lecture)

Class Meetings

MTWR 2:00 PM – 4:20 PM

Barnard Hall 126

- All lectures will be recorded and put on the course website

→ This class can be taken fully remotely

- I will lecture for about 60-90 minutes each day, and remaining time will be “lab time” in Barnard Hall 254

- We will be doing lots of live coding during lecture

- Very similar structure to CSCI 132

Course Logistics (Lab)

- 3:30ish – 4:30ish

Locations: Barnard Hall 254

- Attendance is not required.
- I will be available virtually and in-person to help with any CSCI 232 assignments

Summer Class

- This is a 6-week Summer class. We cover 16-weeks of content in only 6 weeks.
- This class moves fast, and you can expect 1-2 assignments due every week.
- Do not fall behind, and do not slack off, otherwise this class will be tough.
- Be ready to put in several hours a week into this class.
- I am very flexible– I understand people have summer plans and are working parttime/full-time. If you need anything just let me know
- This class can be taken fully online/remotely
- I promise this class won't be miserable

Course Logistics

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

https://www.cs.montana.edu/pearsall/classes/summer2025/232/main.html

CSCI 232: Data Structures and Algorithms

Summer 2025

Quick Links

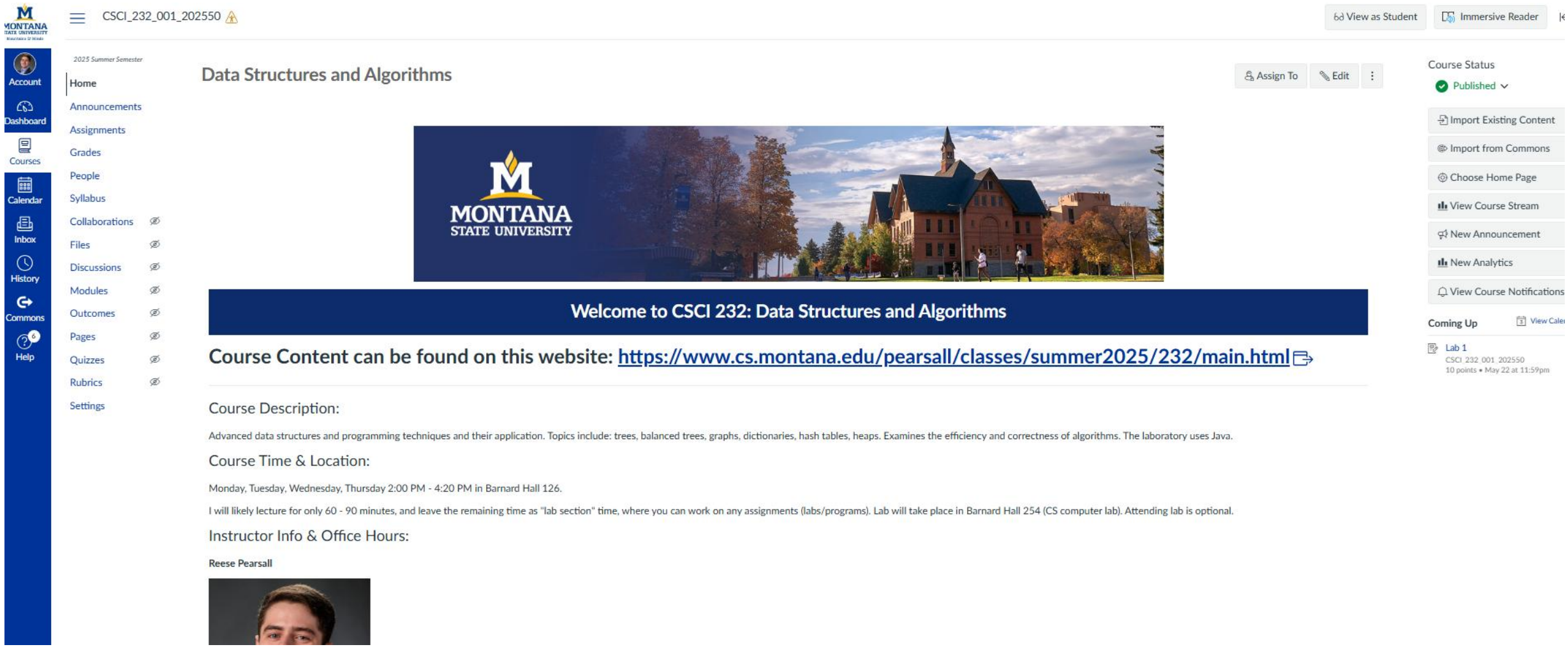
-Syllabus

Date	Topic	Extra Notes	Slides + Lecture Recording	Assignment
Monday May 19th	Syllabus, Java Review	CSCI 132 Material		Please fill out the course questionnaire!
Tuesday May 20th	Data Structures, Time Complexity			
Wednesday May 21st	Trees			
Thursday May 22nd	Trees			
Sunday May 26th				
Monday May 27th	NO CLASS (Memorial Day)			
Tuesday May 28th	Binary Search Trees			
Wednesday May 29th	Binary Search Trees			

Canvas

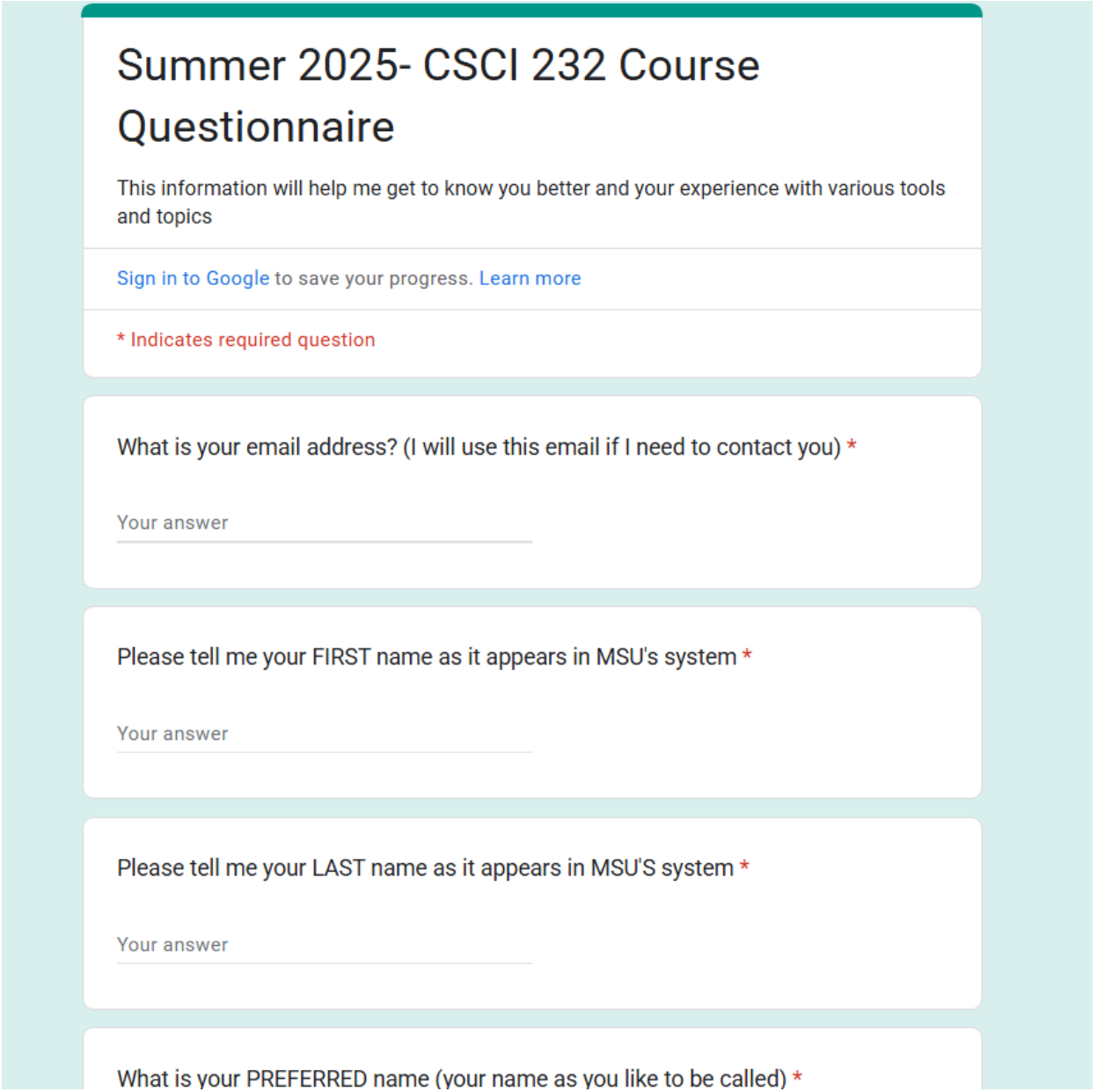
Starting this Summer, **Canvas** is being used as MSU's learning management system

- This is replacing D2L/Brightspace



Course Questionnaire

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



Summer 2025- CSCI 232 Course Questionnaire

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

Sign in to Google to save your progress. Learn more

* 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 system *

Your answer

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

Your answer

What is your PREFERRED name (your name as you like to be called) *

Prerequisites

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

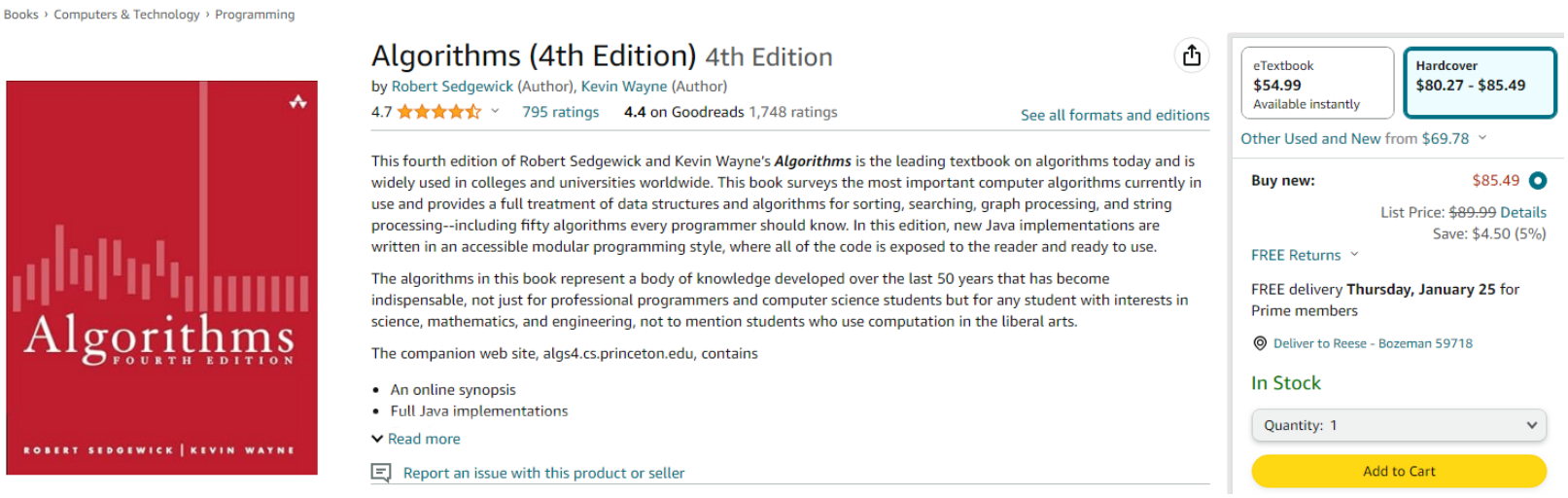
  You will be **totally fine** if you have taken CSCI 246

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.



unfortunately, a very relatable meme

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

Grading

- 30% - Labs (8 @ ~4.3% each)
- 40% - Programs (4 @ 10% each)
- 20% - Quizzes (3 @ 6.67% each)
- 10% - Participation/Check-in

Grading

Labs (30%)

- Shorter, weekly assignments.
- Can generally be finished within 1-2.5 hours
- Due at 1:59 PM on specified days to Canvas (submit .java files)
- I will post the labs a few days ahead of time
- You are usually given starting code
- I will drop your lowest lab grade
- Individual submissions

Grading

Programs (40%)

- Longer, more complicated programming assignments
- Will likely take 2+ hours to complete
- You will always have ~one week to complete them
- Write code from scratch (but we will write some code in class)
- Much higher stakes, make sure you give yourself plenty of time to complete them
- You are allowed to work with 1 partner

Grading

Quizzes (20%)

- There will be three quizzes this semester

- They are taken online through Canvas

- Conceptual questions

→ Multiple Choice
→ True False
→ Matching
→ Short Answer

You can use any notes, past assignments, lecture recordings, slides, documentation, etc

- Should take less than 1 hour.
- Not timed, but only get one attempt
- You can take it from home
- Quizzes will be open 6AM – 11:59 PM on quiz days
- No lecture on quiz days

Grading

Participations/Check-in (10%)

- Helps me make sure you are keeping up and doing ok in the class
- Very easy 10%. You can earn these points in a variety of ways

→ Attend lecture (don't need to attend all of them)
→ Attend "lab" or office hours
→ Schedule a check-in meeting with Reese (office hours or appointment scheduler)

If you are taking this class fully online, you need to meet with me at least once during the summer

Grading Deductions/Late assignments

- 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 Deductions/Late assignments

- 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

If you have a legitimate reason for needing more time on an assignment (travel, work, vacation), just let me know, and I am usually ok with giving you an extension

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

IDE

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

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

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

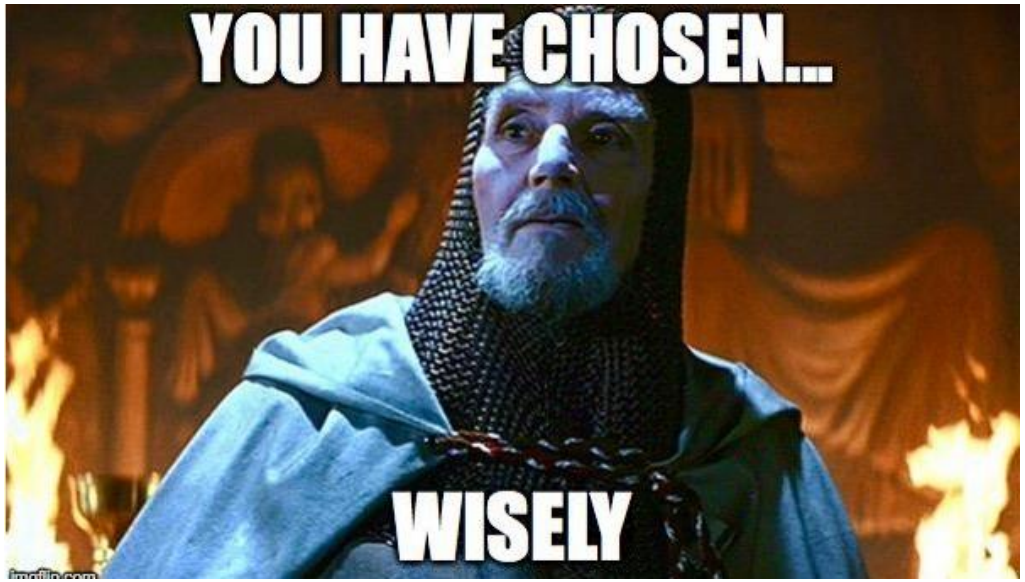
- **Get help when you need it**
- **Keep up with lecture recordings**

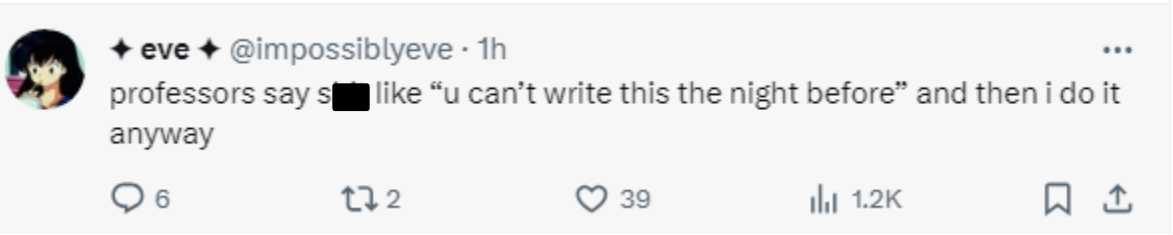
- Get started on assignments early (especially programs)!

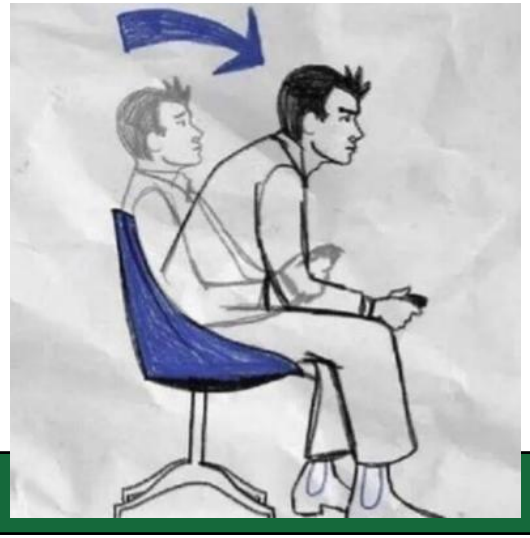
- Come to class and office hours

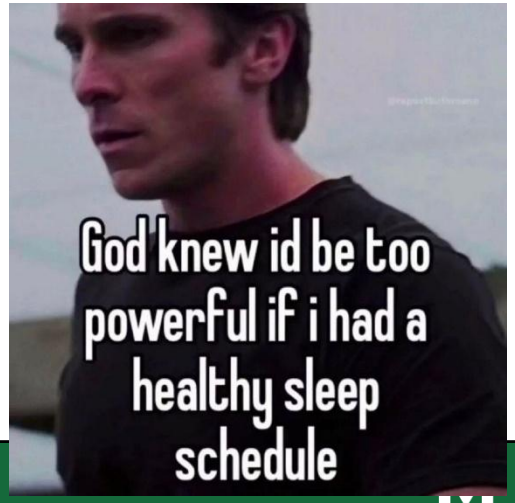
- Take care of yourself

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









Questions?

We are going to write a program where a user can keep track of their online shopping cart.

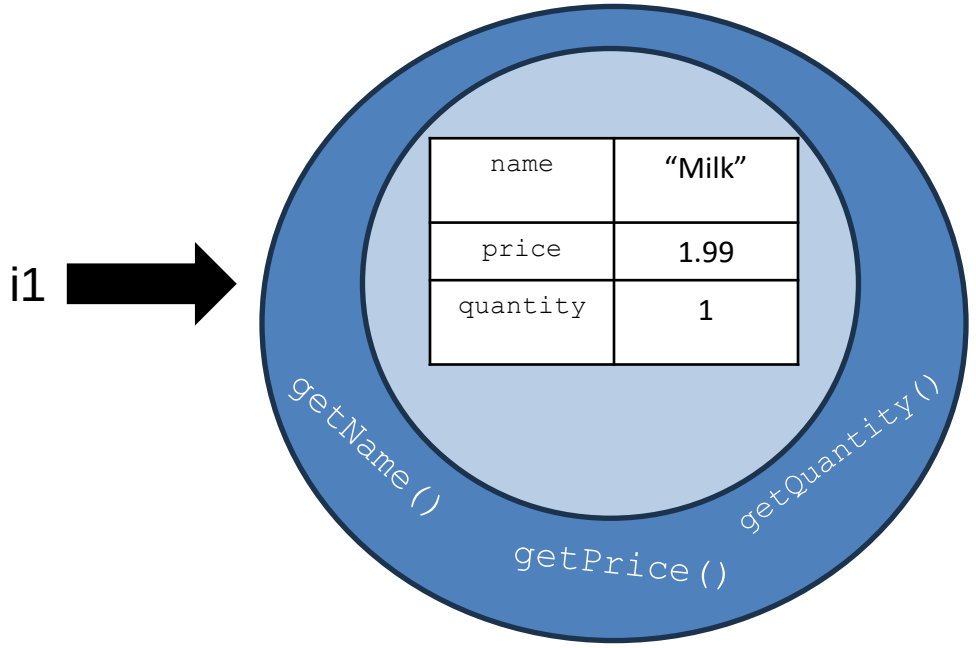
Users can add items, remove items, search for items, get the total price of cart, and apply coupons to items

```java
public class Item {


    private String name;
    private double price;
    private int quantity;

    public Item(String n, double p, int q) {
        this.name = n;
        this.price = p;
        this.quantity = q;
    }

    public String getName() {
        return this.name;
    }

    public double getPrice() {
        return this.price;
    }

    public int getQuantity() {
        return this.quantity;
    }
}
```

Java Class: Blueprint for an object (i.e. a “thing”)

- Instance Field/Attributes
- Methods

```java
Item i1 = new Item("Milk", 1.99, 1);
Item i2 = new Item("Eggs", 3.99, 2);

System.out.println(i1.getName());
System.out.println(i2.getQuantity());
```



Java Objects: **Instances** of classes.
Program entities