Carnegie Mellon ROBOTICS CLUB

Algorithms

Spring 2026

General

- Algorithms refers to the control aspect of robotics
  - How a robot processes the input to solve a problem
- Since the “brain” of a robot is a computer, every computer science algorithm can be run on a robot
- Most of the time, however, we have limited processing power and memory onboard the robot’s computer

Algorithms as a Study

- Taught in the following classes:
  - 15-210, 15-251, 15-451, etc.

- Not necessarily relate to robotics problems, but are still applicable
  - Many 16-xxx (Robotics Institute) courses will teach more applied ones
- Huge field of study and extremely convoluted
- Let’s not get into the specifics...

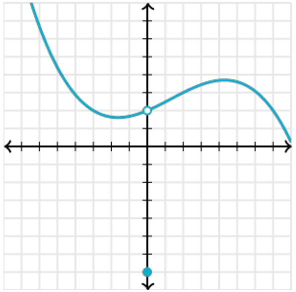
Today's Topic: Graphs

- Converting the real world to something more manageable (read: can be asked on LeetCode):
  - Abstraction of problems and namely graphs!!!!!!!!
- Several types of algorithms depending on the environment:
  - Informed: You know where you are and where to go
  - Uninformed: Everything around you is a mystery
- **Path planning**: wanting to go from A to B
- Also called **graph search** (Searching the graph for the goal)

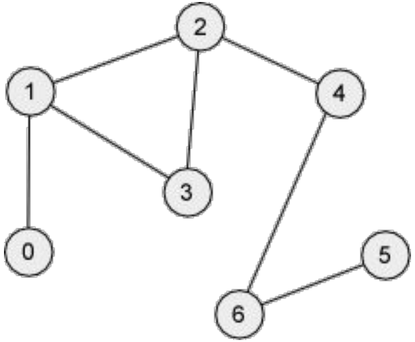
Important Questions

- Why do we need to change real world problems to simple ones?
- How do we choose the way we represent a problem?
- What do we need to retain from a real world problem to a simplified one?
- When do we know when to stop simplifying a problem?

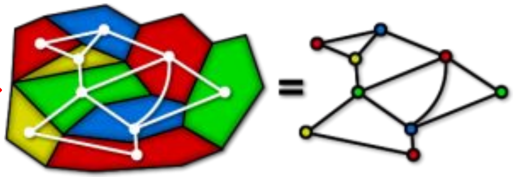
Graphs



Going from this idea...

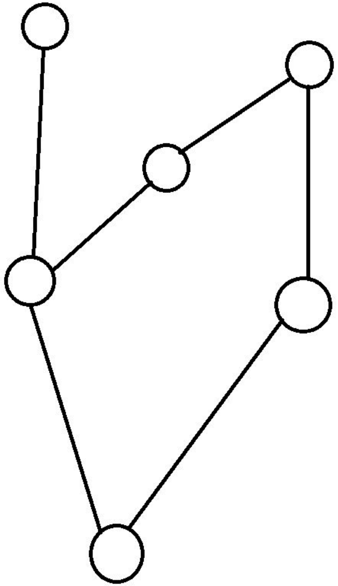


To this idea!



- Graphs are described as a set of points (vertices) connected by lines (edges)
- Due to their simplistic design, they can be used to model a variety of real world environments
- Each vertex can be connected to any number of vertices in the graph through an edge (even itself).
- We use graphs to simplify the real world (a room, a city) and find paths for our robots

Moving Through a City



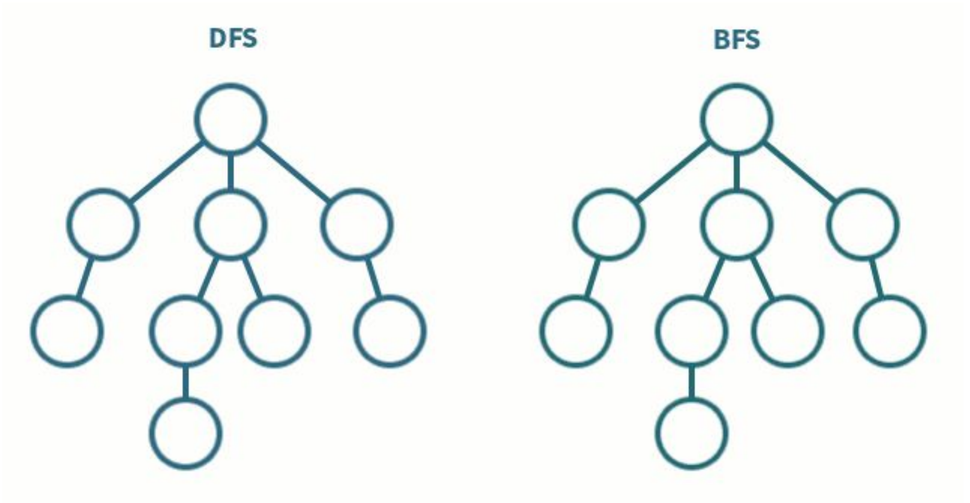
Carnegie Mellon ROBOTICS CLUB

Important Answers

- Why do we need to change real world problems to simple ones?
  - Solving simple problems helps solve more complicated ones
- How do we choose the way we represent a problem?
  - Look at what's needed to solve it
- What do we need to retain from a real world problem to a simplified one?
  - Basic characteristics (variables) and their relationship
- When do we know when to stop simplifying a problem?
  - When doing so changes the nature of the problem or its variables

Uninformed: BFS and DFS

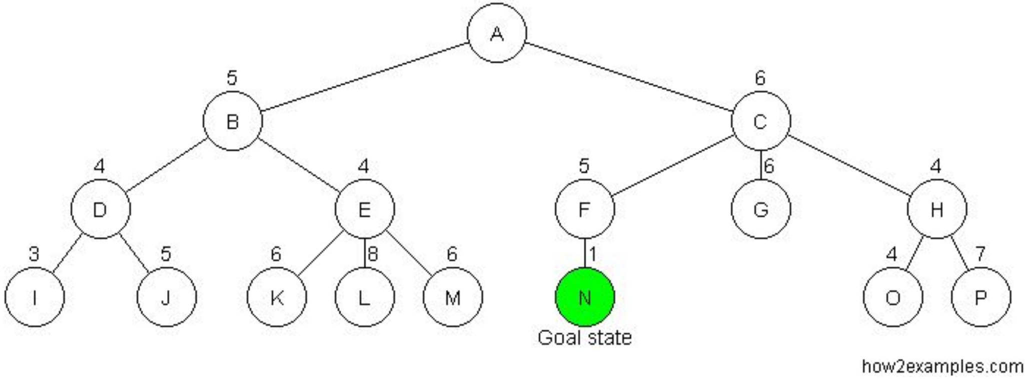
- **DFS(Depth First Search)** follows a path to its end, and then selects the next possible path by backtracking
- **BFS (Breadth First Search)** covers all the closest points available and then moves to the next depth
- They have the same average run time



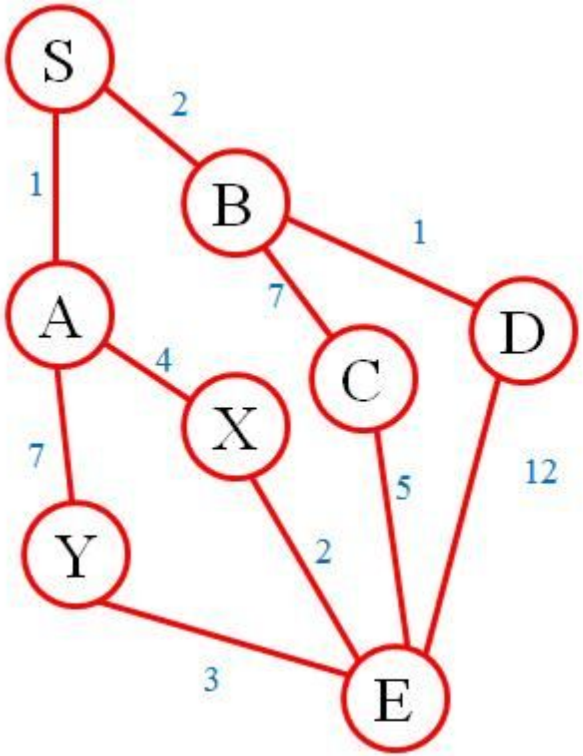
Carnegie Mellon ROBOTICS CLUB

Informed: A* search

- There are cases that you would rather move through a specific path to a point rather than take another (rubble, fire, etc).
- We can model that behaviour by weighing our path and searching through A*.
- A* just looks at every available path and chooses the best one based on a set of rules you set for it (heuristic).



Carnegie Mellon ROBOTICS CLUB



- Values for h:

  A:5, B:6, C:4, D:15, X:5, Y:8

Expand S

{S,A} *f*=1+5=6

{S,B} *f*=2+6=8

Expand A

{S,B} *f*=2+6=8

{S,A,X} *f*=(1+4)+5=10

{S,A,Y} *f*=(1+7)+8=16

Expand B

{S,A,X} *f*=(1+4)+5=10

{S,B,C} *f*= (2+7)+4=13

{S,A,Y} *f*=(1+7)+8=16

{S,B,D} *f*= (2+1)+15=18

Expand X

{S,A,X,E} is the best path... (costing 7)

A* Candy Question

- A* is faster, on average, than uninformed searches, and takes into account information about the environment.
    - Why not just use A* for everything?

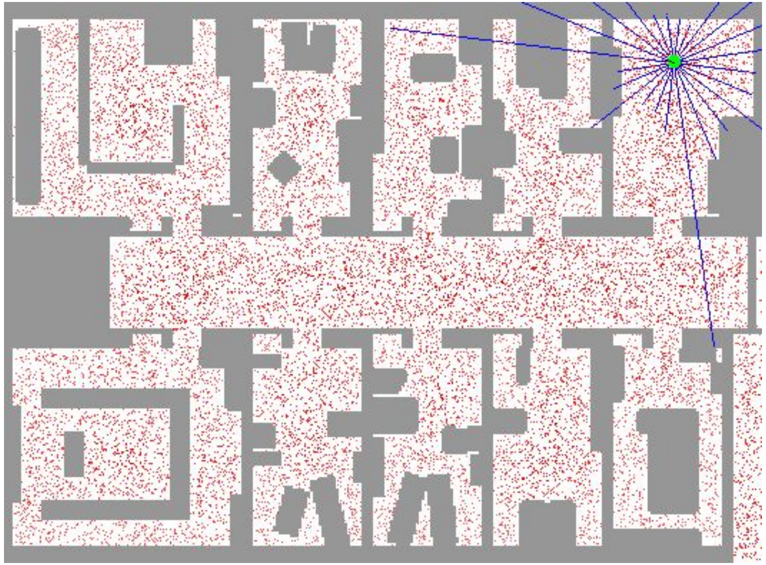
Carnegie Mellon ROBOTICS CLUB

A* Candy Answer!

- A* requires information about the environment and processing power
    - What if you enter a cave with no map?
    - What if you don’t have the processing power?

Localization

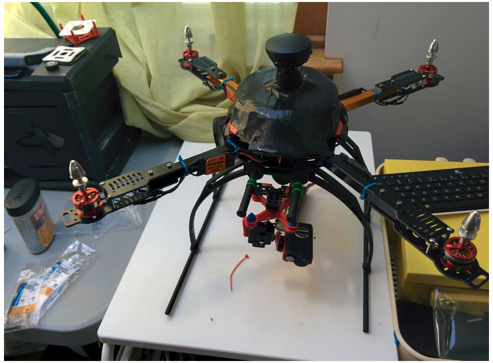
- Sometimes you don’t know where you are in an environment
- You can either know the map and not know your location in it, or have no idea about your surroundings.
- In the former case, you can determine where you are by moving
- This process is called localization



Carnegie Mellon ROBOTICS CLUB

Putting it all together: SLAM

- SLAM lets robots make a map of an unknown environment while keeping track of where they are within that map
- Used to map your environment in relation to self to make a map of your environment as you move



Carnegie Mellon ROBOTICS CLUB

Slam & control in action

Carnegie Mellon ROBOTICS CLUB