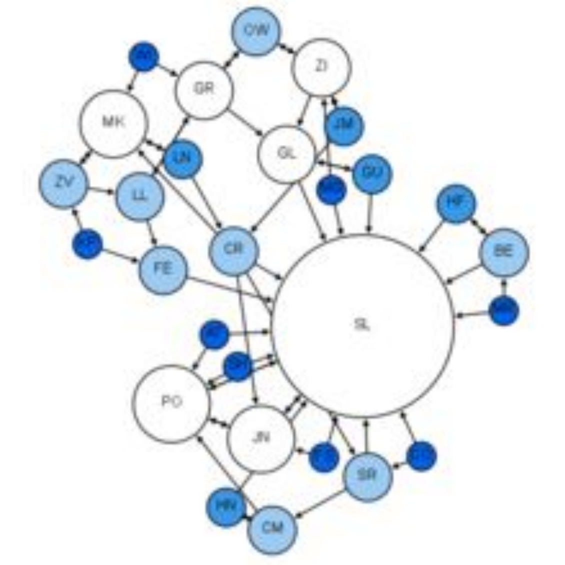
Minimum Spanning Trees

Week 2 Graph Theory AI Inspire Spring 2020

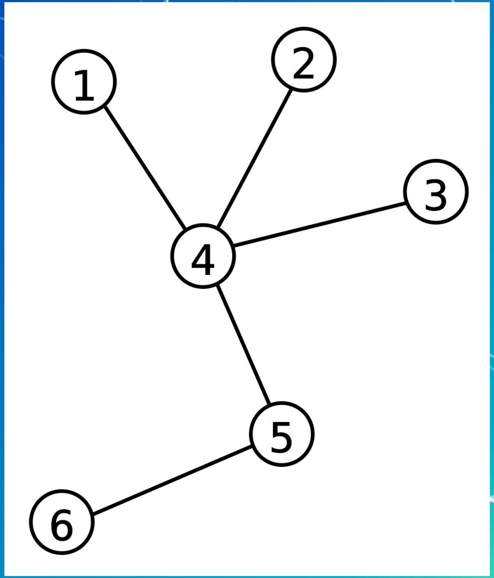
Brief Recap



What is a Graph?
What is a tree?

What is a tree?

- No cycles
- Undirected
- There exists some path between any two nodes in a tree
  - Every node is reachable from every other node

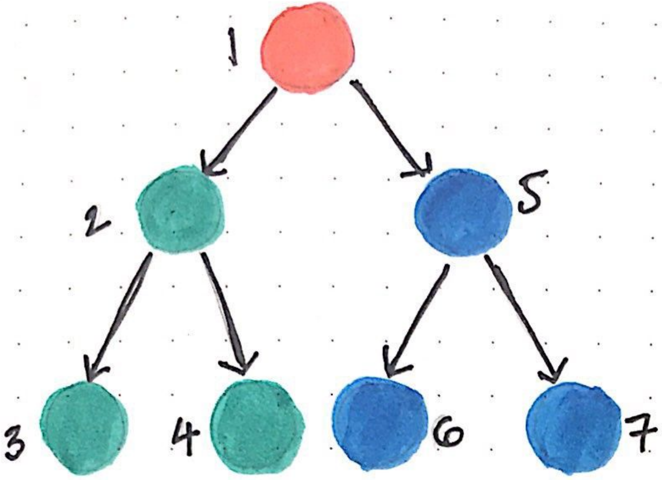


What are some real world applications of graphs?

How can we represent graphs?

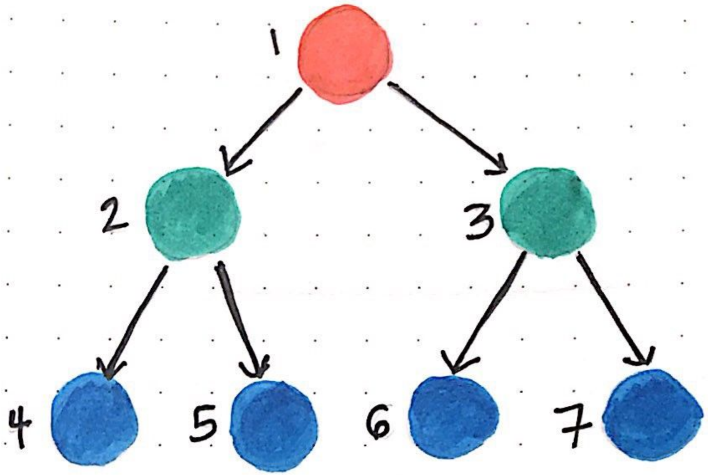
Hint - 2 main ways

What are the two main graph traversals we learned about?



Depth-first search

- Traverse through left subtree(s) first, then traverse through the right subtree(s).

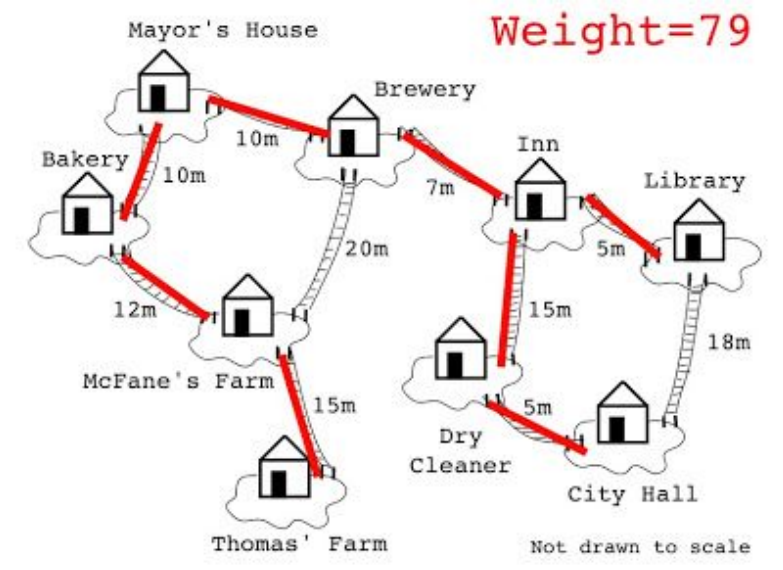


Breadth-first search

- Traverse through one level of children nodes, then traverse through the level of grandchildren nodes (and so on...).

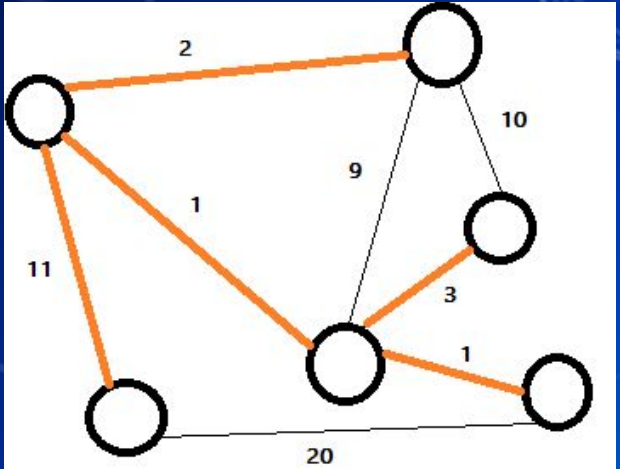
Application/Scenario #1

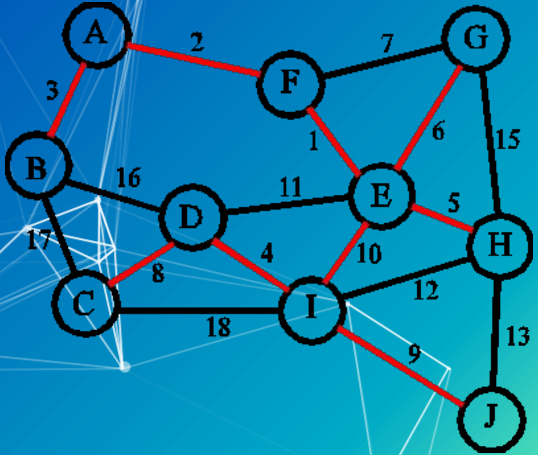
Scenario - Need to lay cables across some town such that all homes are connected but it takes least total cost



Application/Scenario #2 - TSP

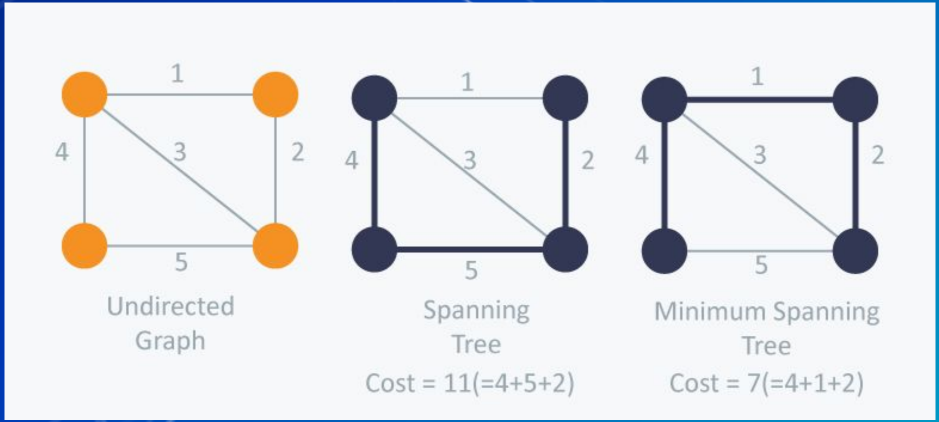
Minimum Spanning Trees





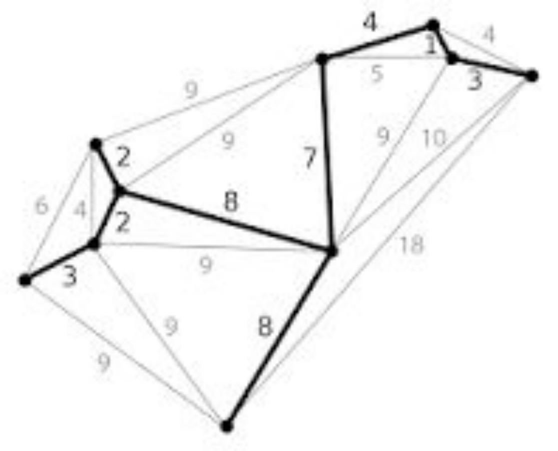
What is a spanning tree?

- Given a set of nodes and edges
- Obtain some type of subgraph s.t. it includes all nodes & is acyclic & it's undirected



What are MSTs?

- Min weight spanning tree
- Given an undirected graph where there are weights on edges ⇒ connect all nodes with MINIMUM poss weight as sum



What are some properties of MSTs?

1) How many edges will they have?
2) If I add 1 more edge, will it create a cycle?
3) If I remove 1 edge, will it disconnect the MST?

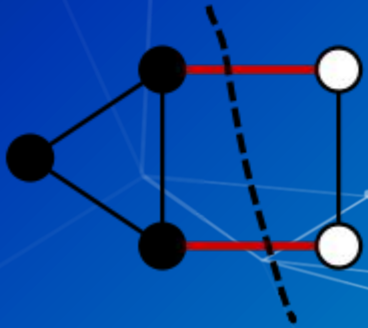
So will a shortest path be obtained through MSTs?

Not always ⇒ but it’s a good approx for Traveling salesman problem

- Hard optimization problem ⇒ Kruskal’s algo which solves MST problem is a good opt.

Cut in Graph

Jmpo property needed for algo to solve MST

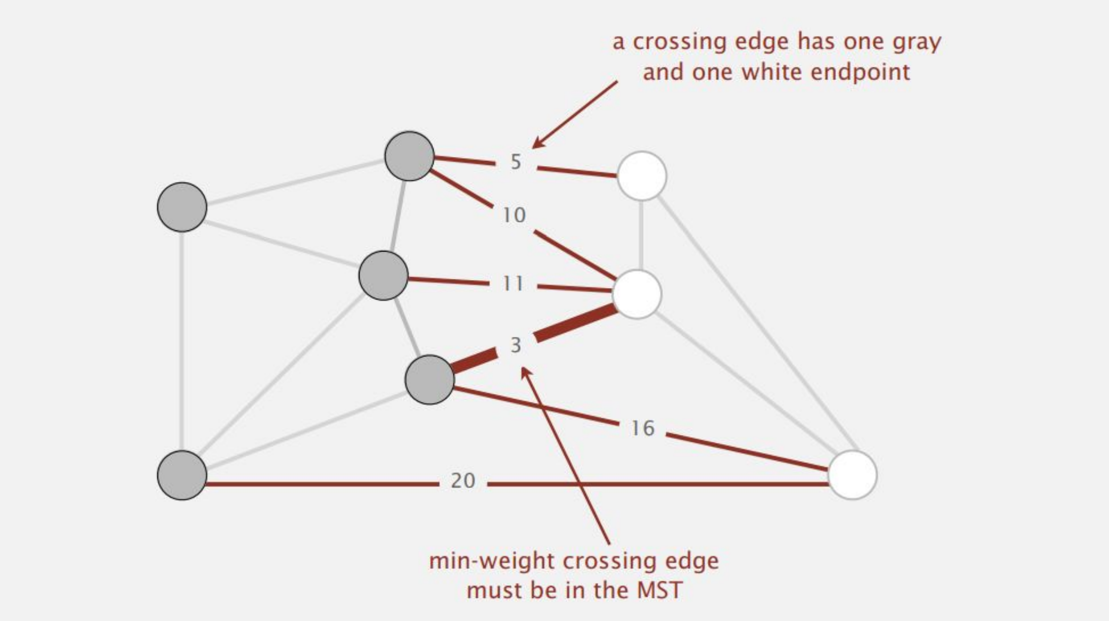


What are some assumptions needed for spanning tree to exist?

- Original graph is connected
- If you want a unique subgraph for the MST
  ⇒ the edges should also be unique

What is a cut?

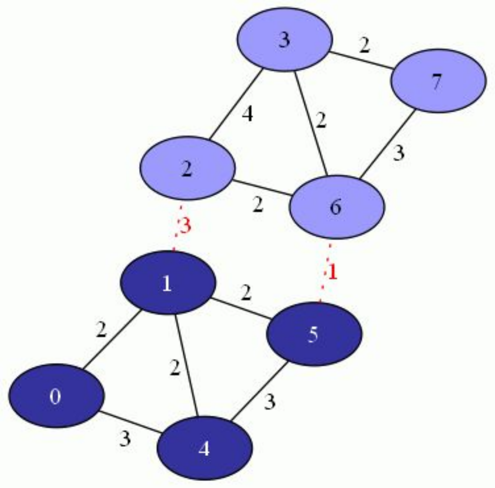
- Cut = partition of nodes into 2 DISJOINT sets
  - Disjoint = separate
- Some edges cross between these two disjoint sets
- Important cut property ⇒ crossing edge with MIN weight is part of MST
  - Helps ensure that we at least know starting point of edges in MST


a crossing edge has one gray and one white endpoint
5
10
11
3
16
20
min-weight crossing edge must be in the MST

Credits = Princeton University (COS 226)

Task

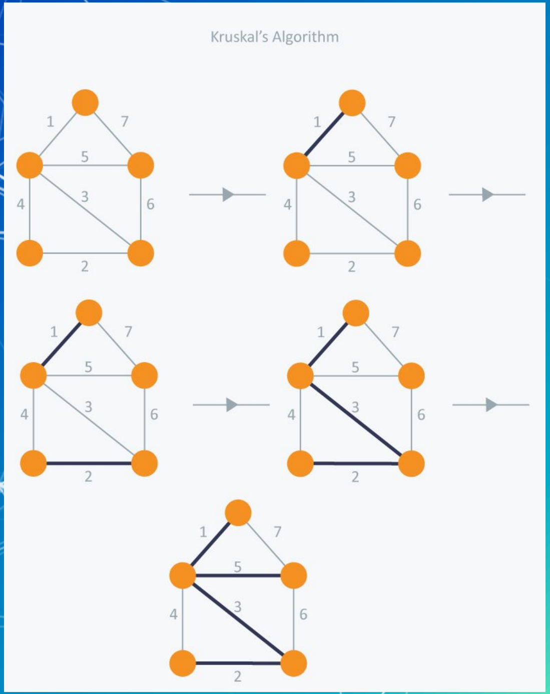
Try proving this cut theorem is correct



How do we solve this MST prob?

2 main popular algos that are known for solving the MST problem

Kruskal's Algorithm

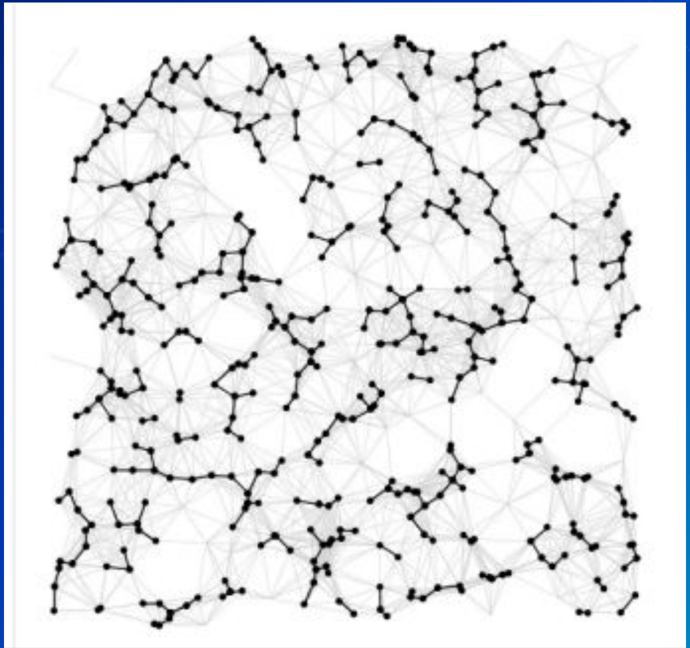


Algorithm

1) Given a graph G with V nodes and E edges
2) Sort the edges in increasing weight
3) Look at the current edge e
    a) Add it to the MST ONLY if it does NOT create a cycle

What type of algorithm does this remind you of?

Kruskal Visualization



Credits = Princeton University (COS 226)

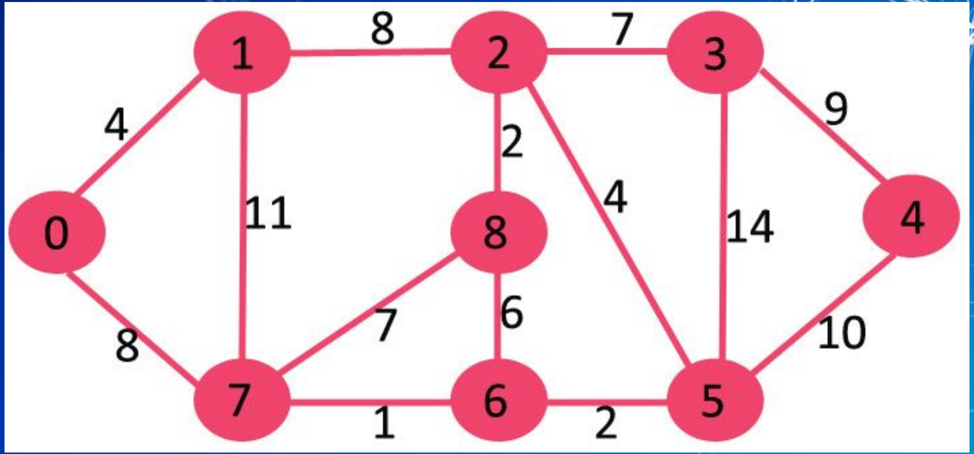
Task

1) How can I find a maximum spanning tree (efficient way)?
    a) Multiple possible solutions
2) How can i find a spanning tree which minimizes sum of cube roots of weights of edges?
    a) Multiple possible solutions

Implementation

- Use Union-find data type
    - We haven’t learned yet ⇒ readings on this data type and how it can be used

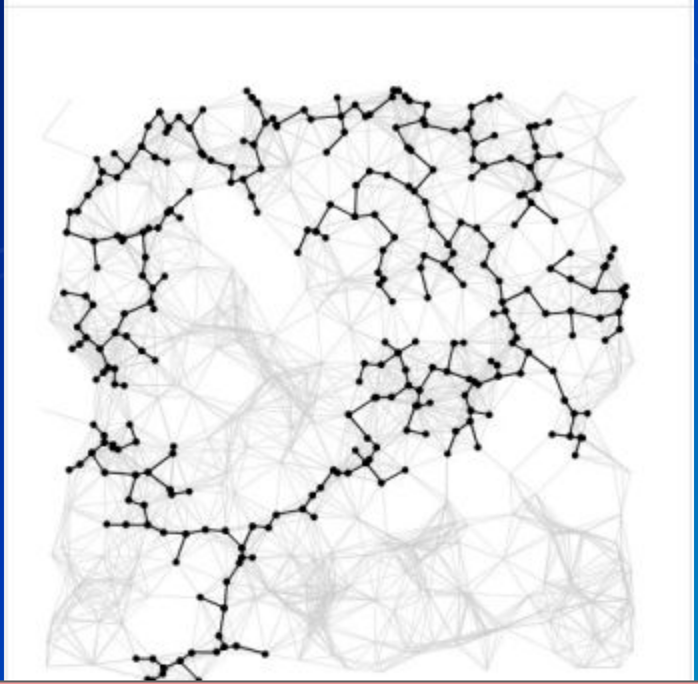
Prim’s Algorithm



Algorithm

1) Given a graph G with V nodes and E edges
2) Start with the a given vertex (generally first one)
3) Until you are done adding edges to MST (V-1 edges) ⇒ add edge with min-weight which has EXACTLY 1 endpoint in the MST subgraph

Visualization



Credits = Princeton University (COS 226)

Implementation

Use a priority queue ⇒ helps sort the edge-weights

Key = edge and priority = edge weight

Always add the edge to the MST with LEAST possible edge weight ⇒ (AKA “delete-min” from PQ binary heap) https://www.youtube.com/watch?v=dNo_BVzNb28

https://www.youtube.com/watch?v=FNJZKLsGAlY

Hope you enjoyed these sessions!