Scientific Computing

Jan 29, 2025

Announcements

→ Office Hours: Mondays + Fridays, 9:30 – 10:30
Cudahy 307

→ HW 1 assigned
On D2L → Dropbox
Due Friday, Jan 31

* Acceptable Sources: Online searches for how to do things in Python cite!

Unacceptable: Searching for the questions, AI Tools

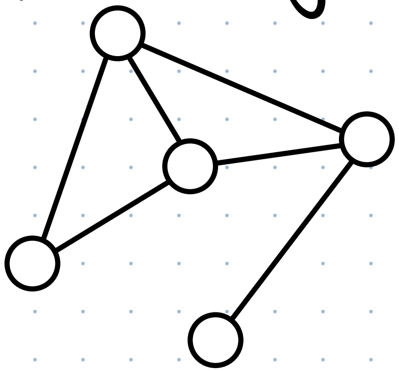
Today

→ Greedy Algorithms

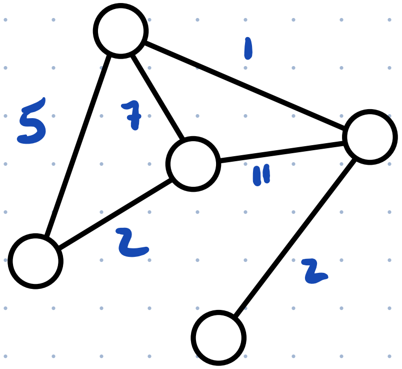
- Coding the greedy algorithm!
    - Python lesson on functions and sort keys
    - Demo

Problem #2: Minimum Spanning Tree

A graph is a set of vertices or nodes, connected in pairs by edges.

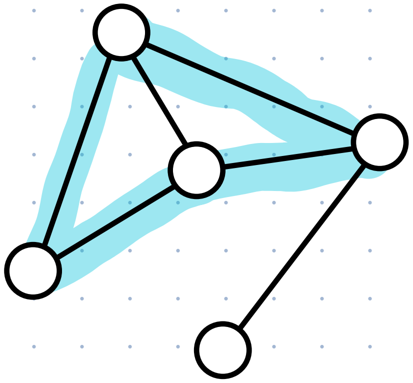


A weighted graph is a graph whose edges have real #s as "weights".

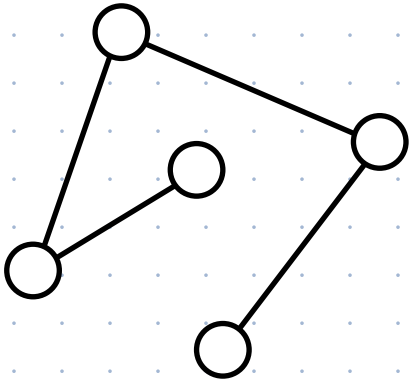


can reach every vertex from every vertex

A tree is a graph that is connected and has no cycles.

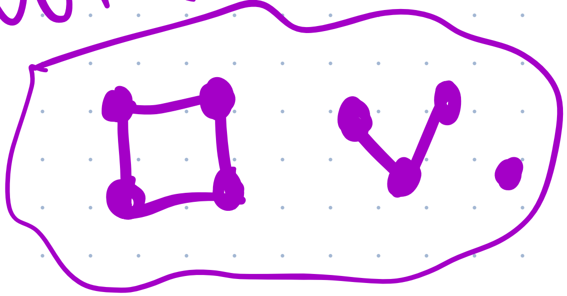


lots of cycles



no cycles = tree

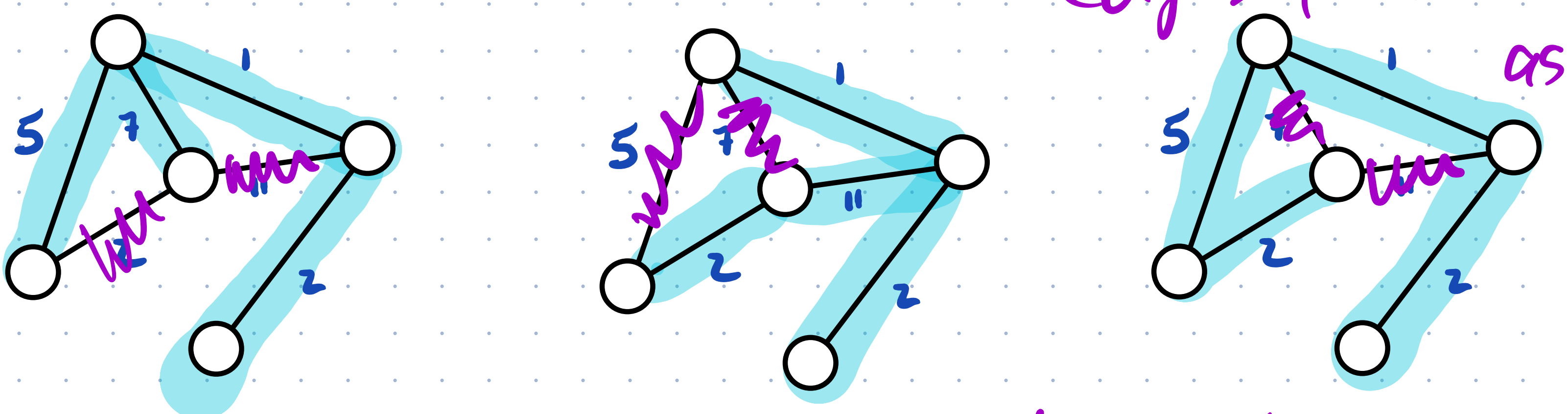
Not connected



Minimum Spanning Tree Problem:

Given a weighted graph G, find the subset of edges that forms a minimum-weight spanning tree.

get a tree by deleting edges, with as small as possible sum of weights



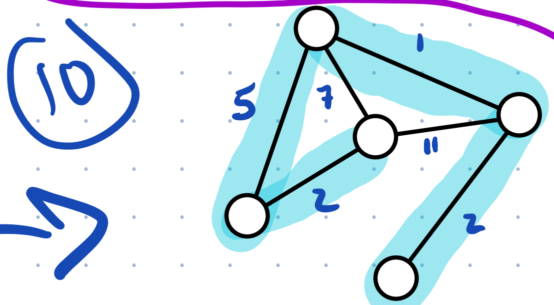
$5+7+1+2=15$ $1+2+2+11=16$ $5+2+1+2=\boxed{10}$

minimum possible!

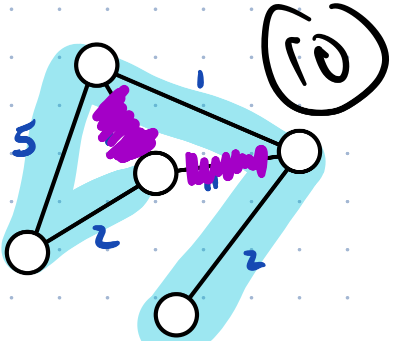
Ex: You might need to connect a bunch of buildings with cables, and the weight of the edges is the cost of the connection.

Possible Greedy Algorithms:

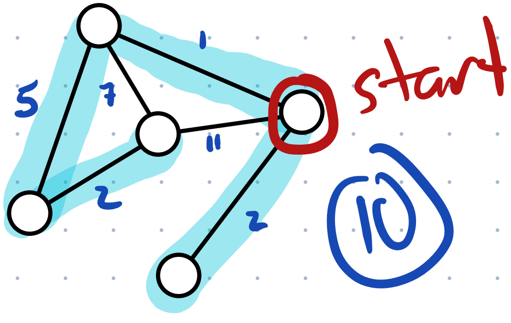
* pick the cheapest edge that doesn't make a cycle
* start with all edges, and delete the most expensive one as long as it doesn't disconnect the graph
* pick one node as the start, and repeatedly choose the cheapest edge that connects to a node you have reached so far



Idea #1



Idea #2



Idea #3

In this example they all generated the same tree, but that doesn't always have to be true.

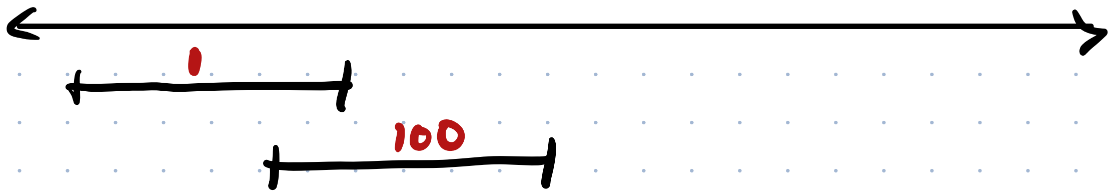
More importantly: are any of these guaranteed to give optimal solutions?

<u>Theorem</u>: They all do!
(We won't prove it in class.)

Problem #3: Weighted Interval Scheduling

This is like regular interval scheduling, except each request i comes with a value v_i and your goal is to maximize the total value of satisfied requests.

Our previous greedy algorithm is now pretty bad.



Maximizing!

Possible Greedy Algos:

- best = most profitable (highest weight)
- best = shortest
- best = least conflicting
- best = [of the 10 shortest meetings, the most profitable]
- best = ______________________________

[if ties, go earliest end time]

Are any of these optimal?

[demo]