Feb 5, 2024

Greedy Algorithms (continued)

Announcements:

- → HW 1 due on Friday
- → Normal Office Hours this week

Negative indexing:

L a list

L[-1] = the last thing in the list

```
# sort L
L.sort()
```

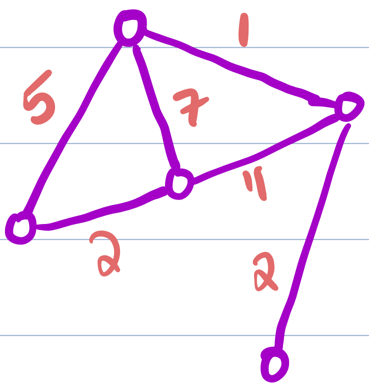
L = ["a", "b", "c", "d"]

L[0] = "a"   L[1] = "b"   L[2] = "c"   L[3] = "d"

L[-1] = "d"   L[-2] = "c"   . . .

Example # 2: Minimum Spanning Tree

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



5 vertices, 6 edges

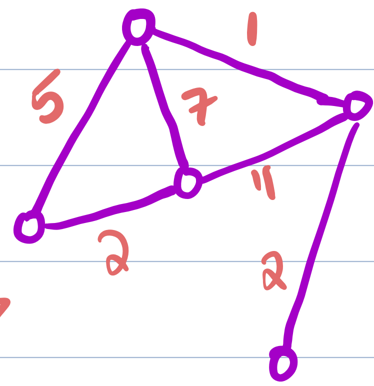
Def: A <u>weighted graph</u> is a graph in which each edge has a real # as its "weight."

Def: A <u>tree</u> is a graph that is <u>connected</u> and has no <u>cycles</u>.

connected ↙ can reach any vertex from any other vertex by traveling along edges

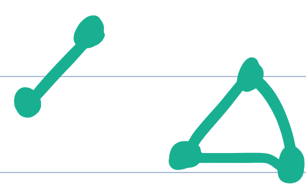
cycle = loop of edges

a path from a vertex back to itself without using the same edge twice



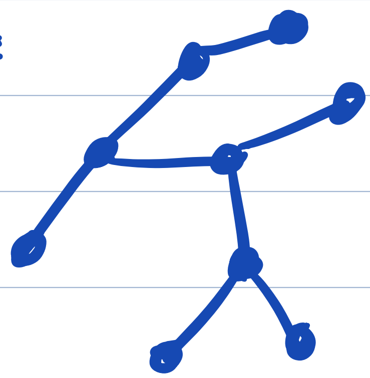
connected ✓

lots of cycles



not connected ✗

Tree:

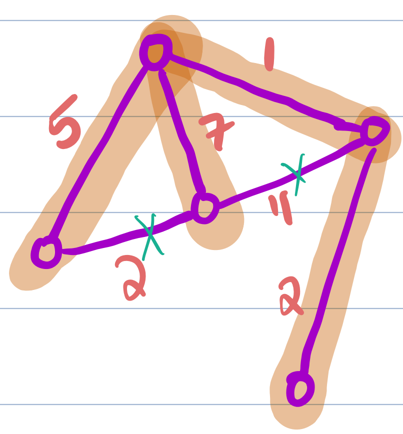


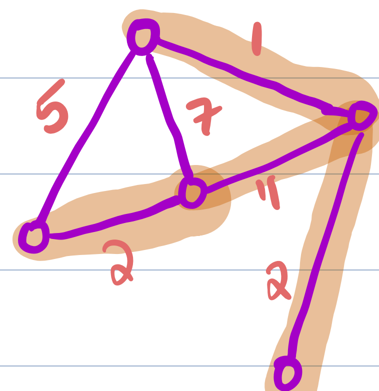
connected ✓
no cycles ✓

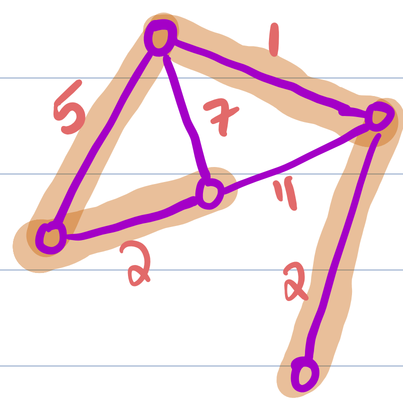
Minimum Spanning Tree Problem:

Given a weighted graph G, find the subset of its edges that forms a spanning tree and the sum of its edges is as small as possible.

a tree on the graph that uses only existing edges







$2+11+1+2=16$

$5+7+1+2=15$

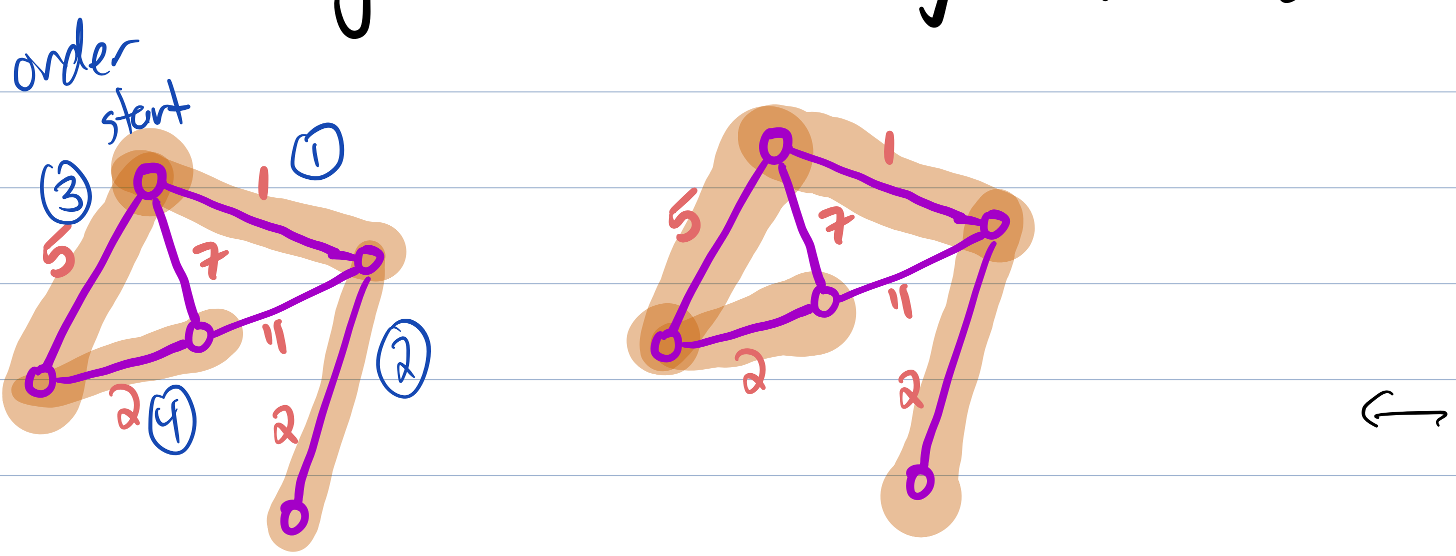
$5+2+2+1=10$
minimum possible

Ex: The vertices are buildings on campus
Edges = buildings that can be connected by a fiber optic cable and the cost

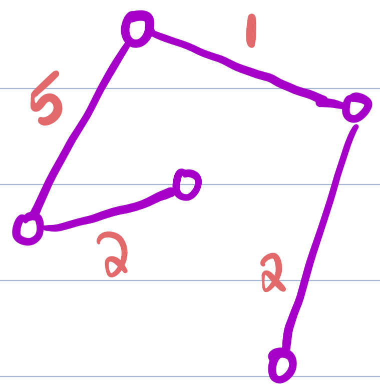
Minimal Spanning Tree: The cheapest way to network all the buildings together.

Possible greedy algorithms:

* pick one node as the start and repeatedly choose the cheapest edge that connects to a node that you're not already connected to.

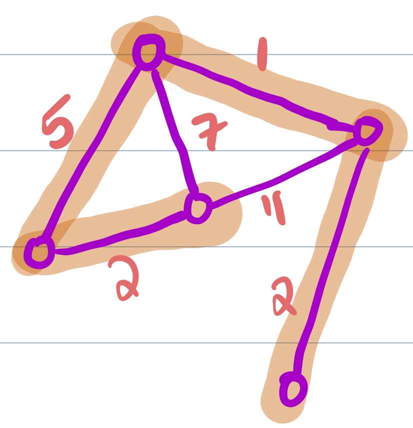


* start all edges, and repeatedly the most expensive one that does not disconnect the graph



done

* start with no edges and repeatedly add the cheapest one that does not create a cycle



done, same solution

* In this small example, all three greedy algos. gave the same solution, but

that's not always the case.

<u>Important question</u>: Is it guaranteed to be optimal?

<u>Theorem</u>: All three of these GAs are optimal.