Lecture 22: Minimum spanning trees

- MST Definition
- Prim's & kruskal's algorithm
- Project 5 overview + software engineering

Announcements:

- P5 extension → Wednesday

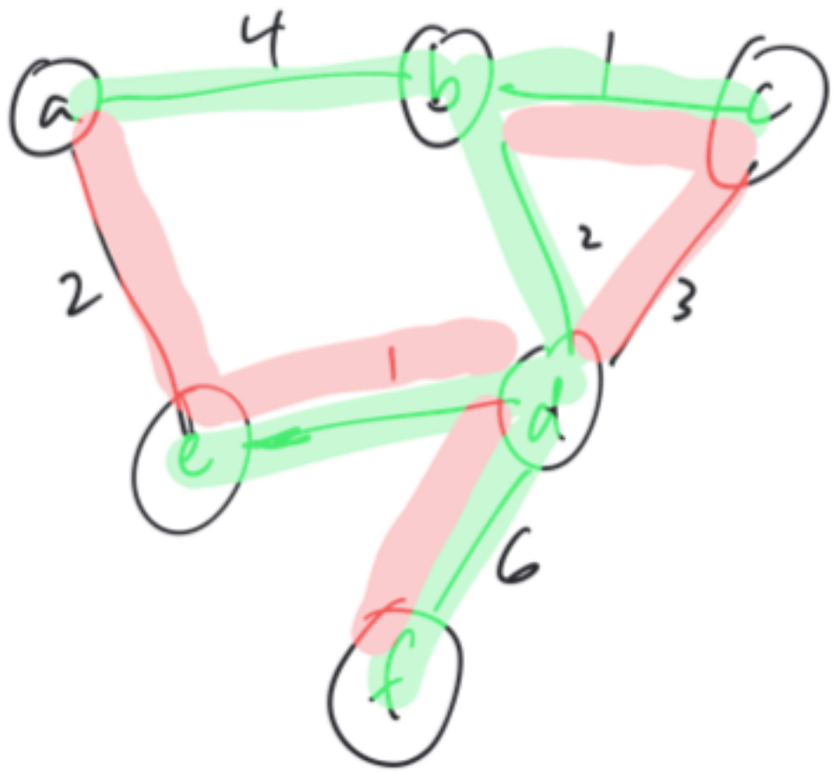
A <u>tree</u> $T=(V',E')$ in a graph $G=(V,E)$ is a subgraph of G satisfying:

1. T is connected
2. T is acyclic
3. $|E'|=|V'|-1$

Note: any 2 of these imply the third.

A <u>spanning tree</u> is a tree in G that has all of G's vertices (i.e $V'=V$)

Example: phone tree.



total cost:

$4+1+2+1+6=14$

total cost

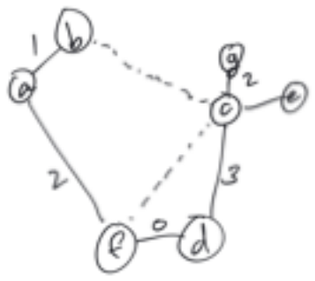
$2+1+1+3+6=13$

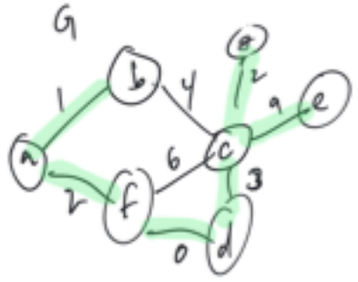
(MST) A minimal spanning tree T in a graph G is a spanning tree with minimal total weight along all edges in T.

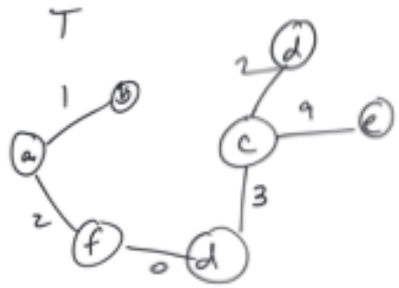
Finding a MST:
Idea: maintain a subgraph T, keep adding edges until it is a ST.
- choose edges to add in a way that gives Minimal spanning tree.

Prim's algorithm:
invariant: T is the smallest (total weight) tree (connected) that contains m nodes & start vertex.

Kruskal's:
invariant: T is the smallest forest in the graph with m nodes
— always add smallest edge that doesn't create a cycle.







To find maximum of a continuous f^n
greedy: always go uphill.

greedy algorithm: at each step, select next 'best' option { always add the edge leading from T with minimum weight.

— if you can show they work, often efficient.