

Prim's algorithm

Goal: to find min spanning tree of a graph

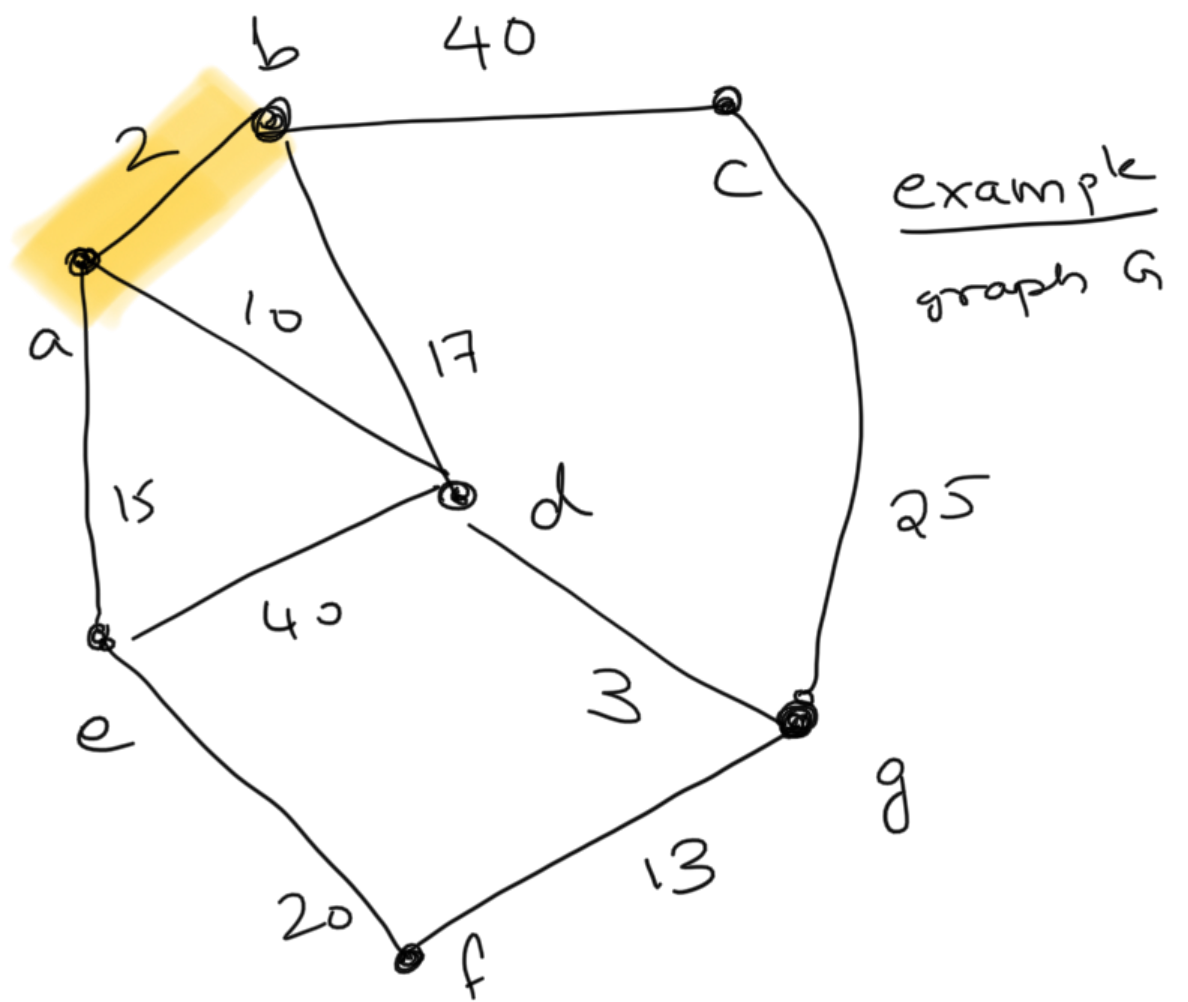
Idea: "grow the tree"

Algorithm: V = vertices of graph
 S = vertices of growing tree
 T = edges of growing tree

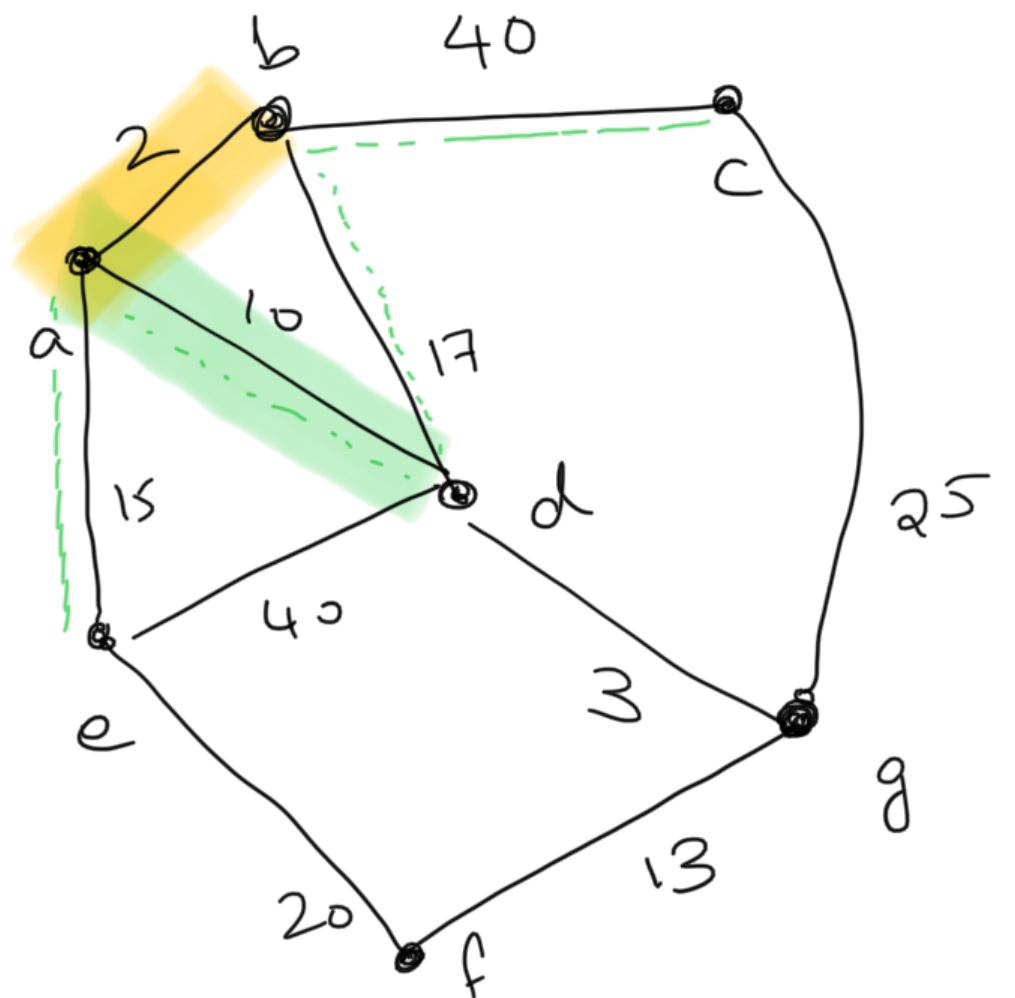
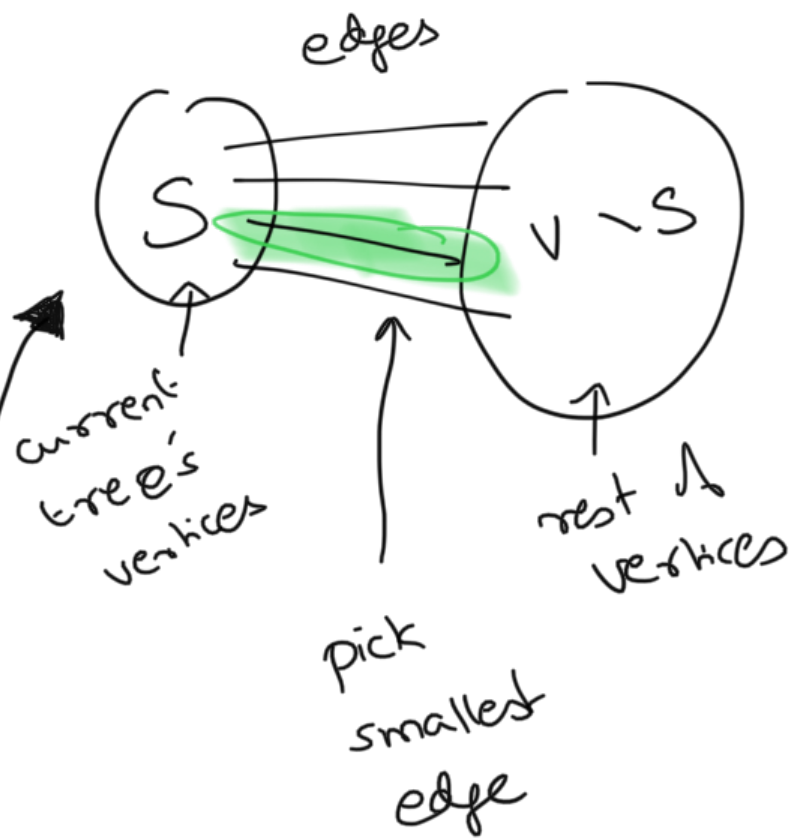
Initial step
I) Pick smallest edge

$S = \{a, b\}$

$T = \{ \text{edge } a-b \}$



II)



$S = \{a, b, d\}$

$T = \{ \text{a-b}, \text{a-d} \}$

out of edges

b-c 40
b-d 17
a-d 10
a-e 15

Picked a-d 10

III)

Repeat step II until $V - S = \emptyset$

Complexity

→ At each step, add vertex

→ so need $|V|$ iterations

In each iteration, have to pick

least edge among at most $|E|$ edges

→ so $|E|$

so $O(|V| |E|)$

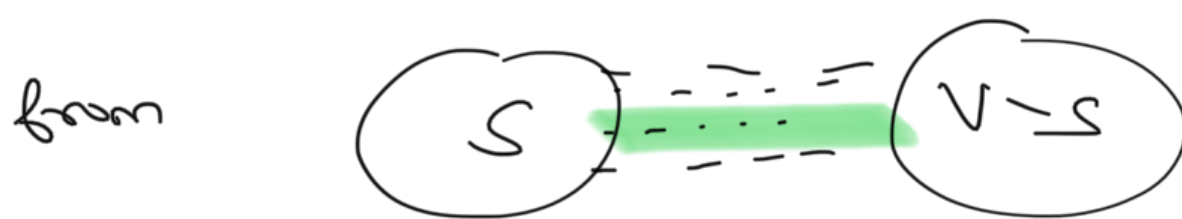
$V = \text{vertices of } G$

$E = \text{edges of } G$

Better implementation?

Use "priority queue" to

efficiently pick smallest edge



- what is a priority queue?
- how to implement it?
- what is the complexity analysis of Prim's in that case?

For next time