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

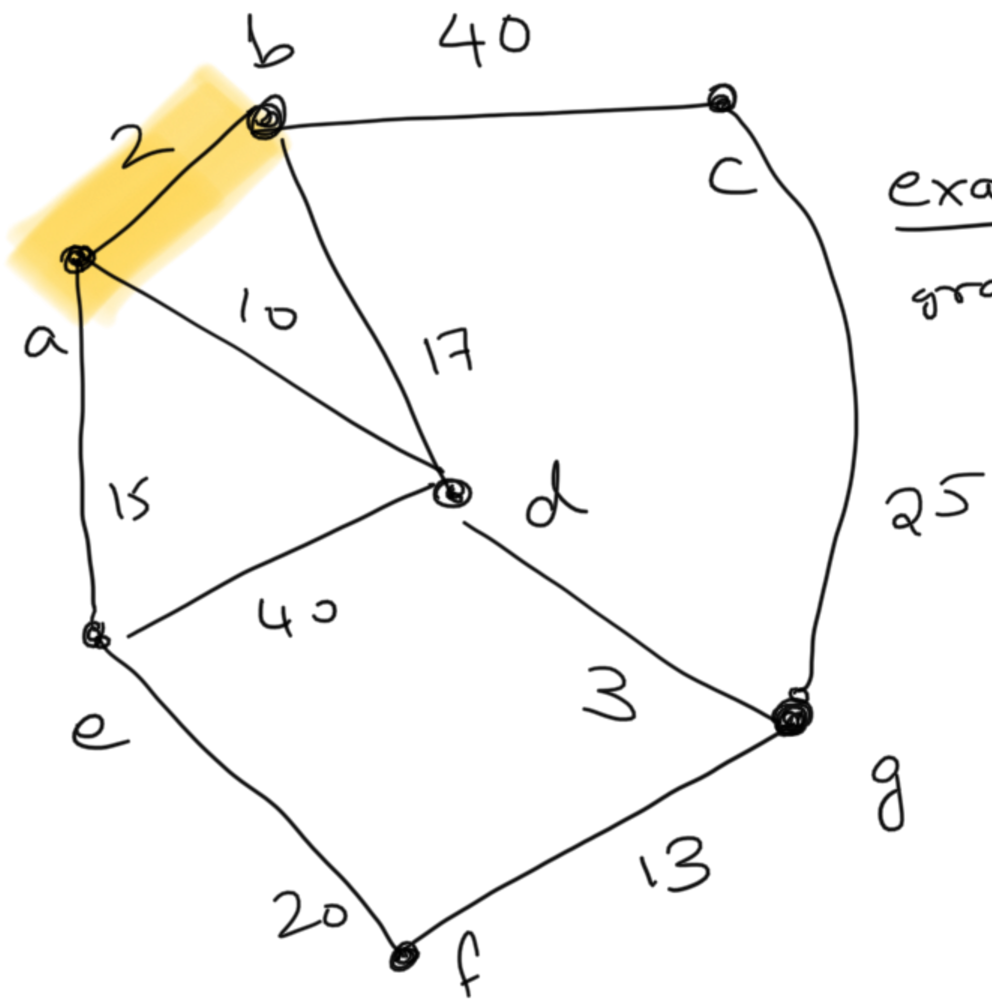
T = edges of growing tree

Initial step

I) Pick smallest edge

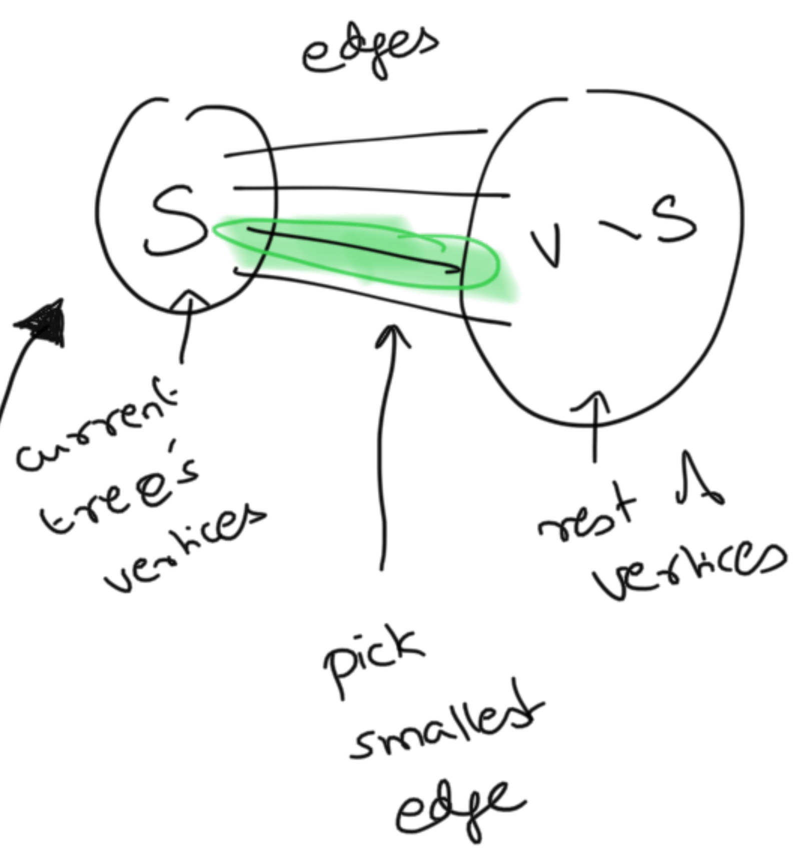
S = { a, b }

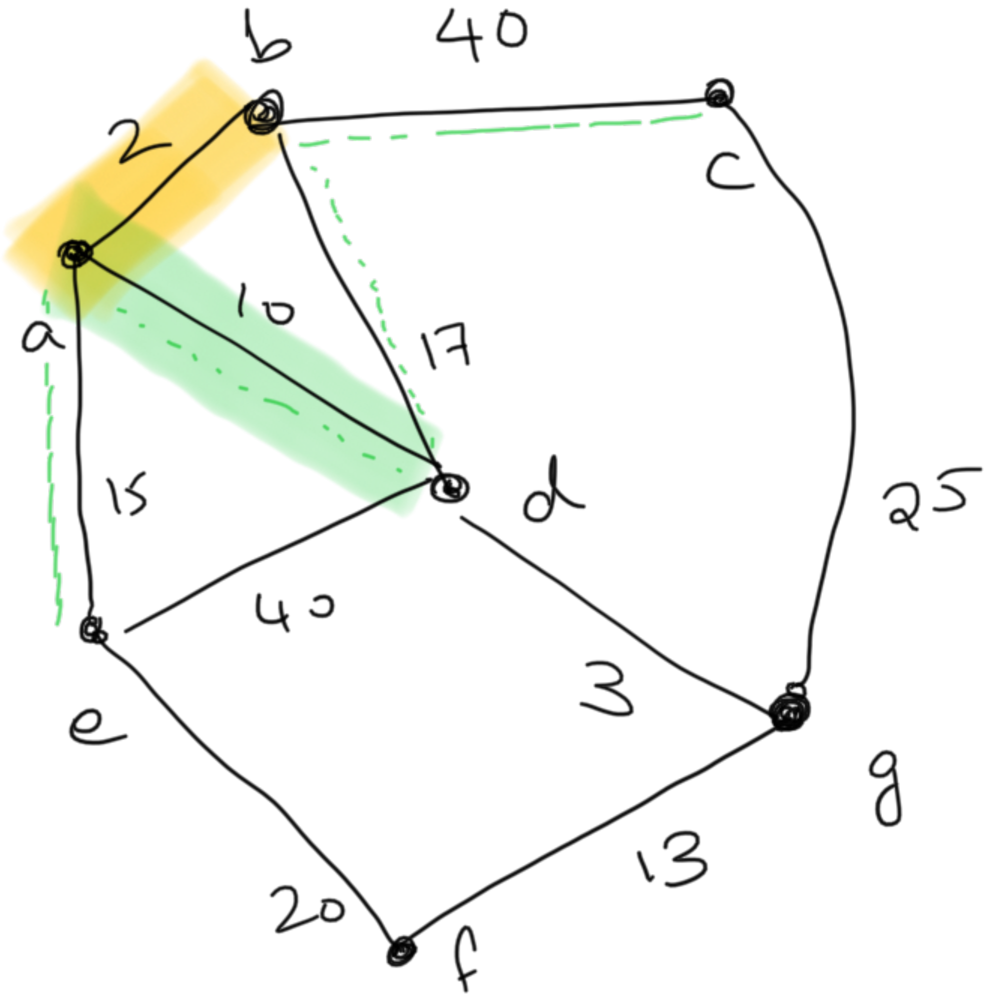
T = { a —— b }



example graph G

II)





S = { a, b, d }

T = { a—b , 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 $V \setminus S = \emptyset$

Complexity

→ At each step, add vertex

→ so need $|V|$ iterations

In each iteration, have to pick least edge among atmost $|E|$ edges

→ so $|E|$

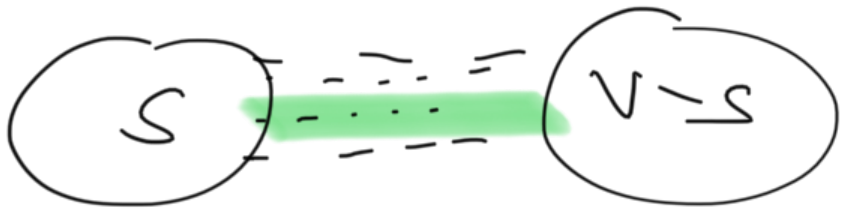
V = vertices of G

E = edges of G

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

Better implementation?

Use " priority queue " to efficiently pick smallest edge from



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

For next time