Computer Algorithms I

Spring 2020

Minimum spanning trees (MST)

build roads connecting every city as cheaply as possible!

$G = (V, E)$ undirected graph, weighted edges: $w(u, v)$

connected graph: there is a path between any two vertices

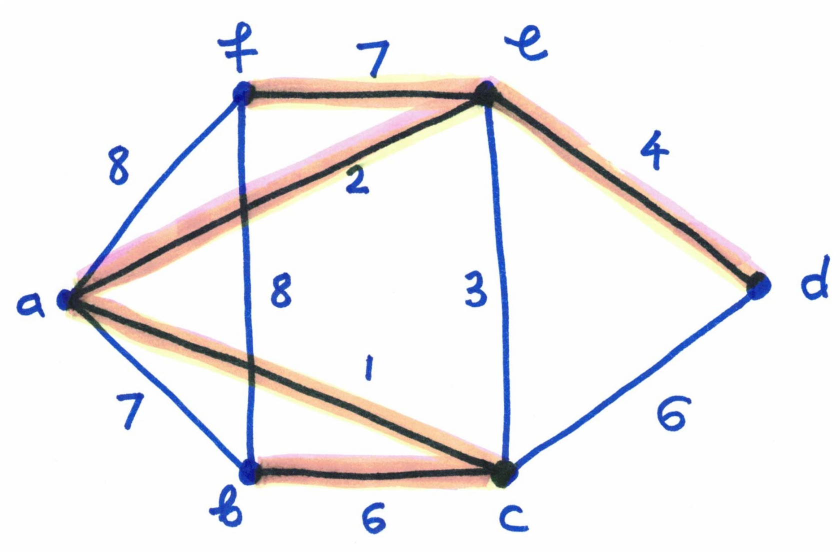
subgraph of G: $G' = (V', E')$ such that $V' \subseteq V$, $E' \subseteq E$

tree: connected graph without cycles

spanning tree of G: tree subgraph containing all the vertices

given a connected, weighted graph, find a spanning tree such that the total weight of edges is as small as possible!

Example



Example

```
MST-PRIM(G, w, r)
for each u ∈ G.V
    u.key = ∞
    u.π = NIL
r.key = 0
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    for each v ∈ G.Adj[u]
        if v ∈ Q and w(u, v) < v.key
            v.π = u
            v.key = w(u, v)
```

Q: priority queue of vertices not in current tree

$(u.\pi, u)$: minimum weight edge between u and current tree

$u.key = w(u.\pi, u)$

tree edges: $(u.\pi, u)$ for $u \notin Q$

Complexity and correctness

looks like BFS, DFS: $O(|V| + |E|)$, but

line 7 is executed $|V|$ times: Extract-Min operation: $O(\log |V|)$

lines 9-11 are executed $|E|$ times: Decrease-Key operation: $O(\log |V|)$

so complexity is $O(|E| \cdot \log |V|)$

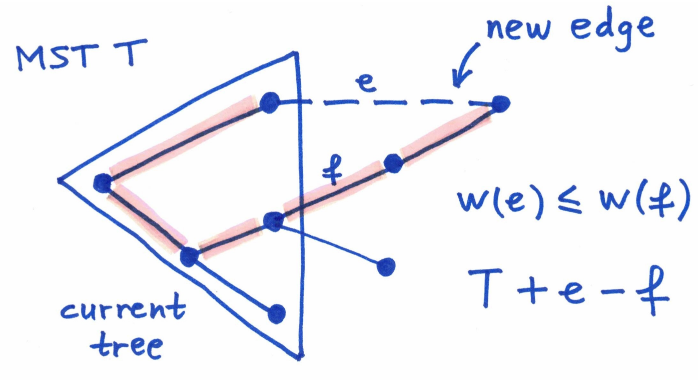
correctness: this is a greedy algorithm, argue as for activity selection!

Claim: after each iteration the current tree is contained in an MST

if next edge is not in MST containing current tree, show that MST can be changed to include next edge

Correctness proof

Claim: after each iteration the current tree is contained in a MST



note: book has more general form of the argument