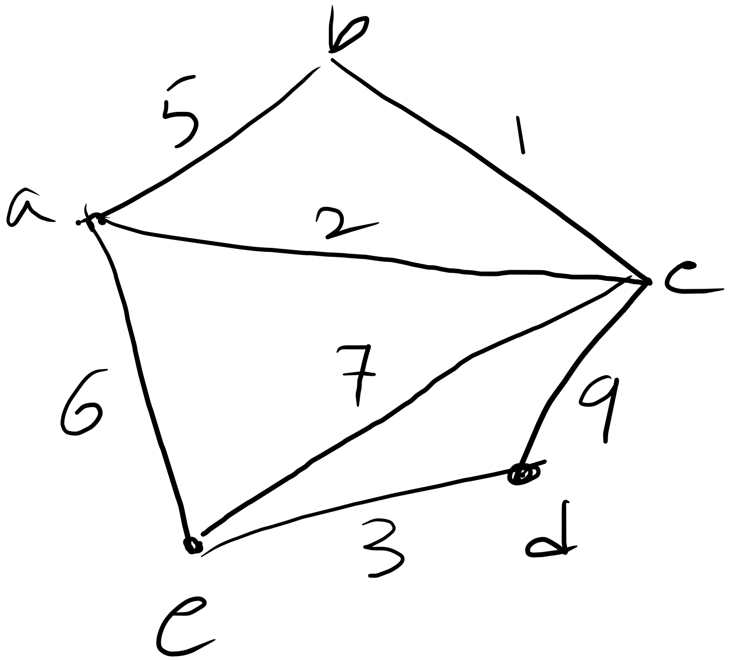
Minimum spanning tree (MST)

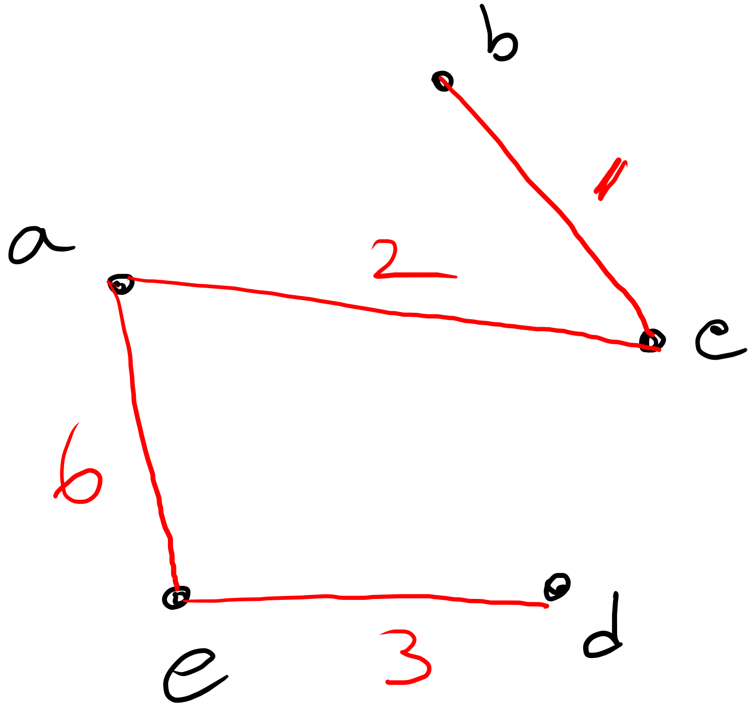
Problem: A connected graph $G(V, E)$ with edge costs.

$$w : E \to \mathbb{R}^+$$

Feasible solution: A subset $T \subseteq E$ such that $G_T(V, T)$ is connected

objective Total cost of all the edges in T is minimum.





cost of $T = 12$

Greedy algorithm

Generic-MST (G, w)

T is empty

While E is not empty

choose e in E

if (e satisfies some condition)

add e to T

Return the set T

The graph $G_T(V, T)$ is ~~the~~ MST.

main task: → In what order should the edges be processed?

→ when should an edge be a part of the spanning tree?

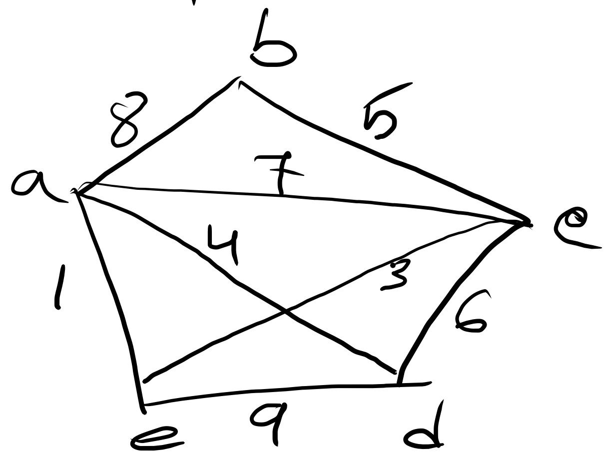
Kruskal's algorithm

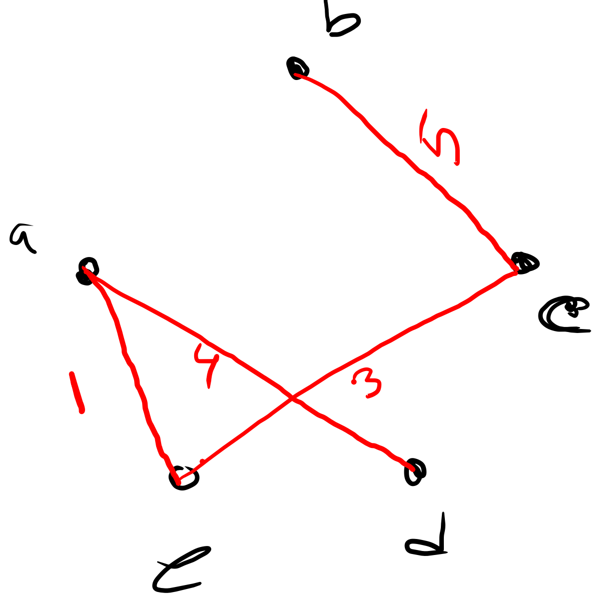
Kruskal-MST (G, w)

- T is empty
- sort the edges of G in non-decreasing order by their weights
- while E is nonempty
- choose e ∈ E in the above order
- if (e does not form a cycle with the edges in T)
- add e to T
- return the set T.

1 3 4 5 6 7 8 9

T = {(a,e), (c,e), (a,d), (b,c),





cost = 13

Prim's algorithm

Prim-MST(G, w)

- T is empty
- $X = \{s\}$ where s is an arbitrary start vertex.
- while there is an edge $e = (u, v)$ where $u \in X$ and $v \notin X$

  $e^* = (u^*, v^*)$ be a minimum such edge

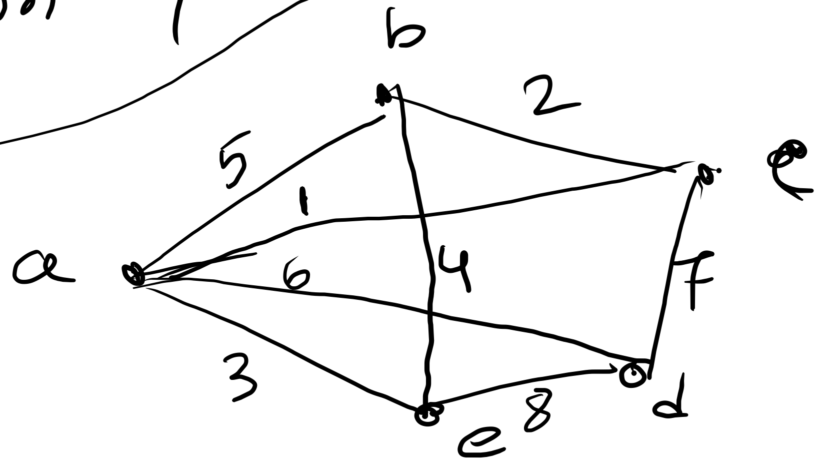
  add v^* in X

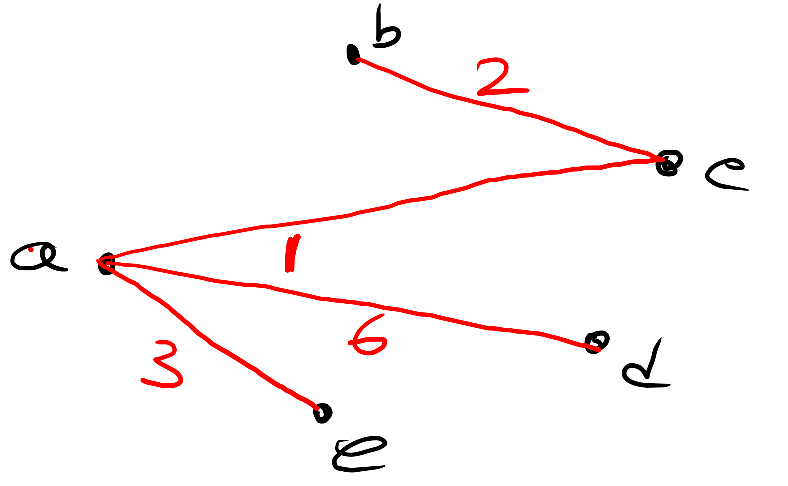
  add e^* in T

return T

$T = \{(a,c), (c,b), (a,e), (a,d)\}$

$X = \{a, c, b, e, d\}$

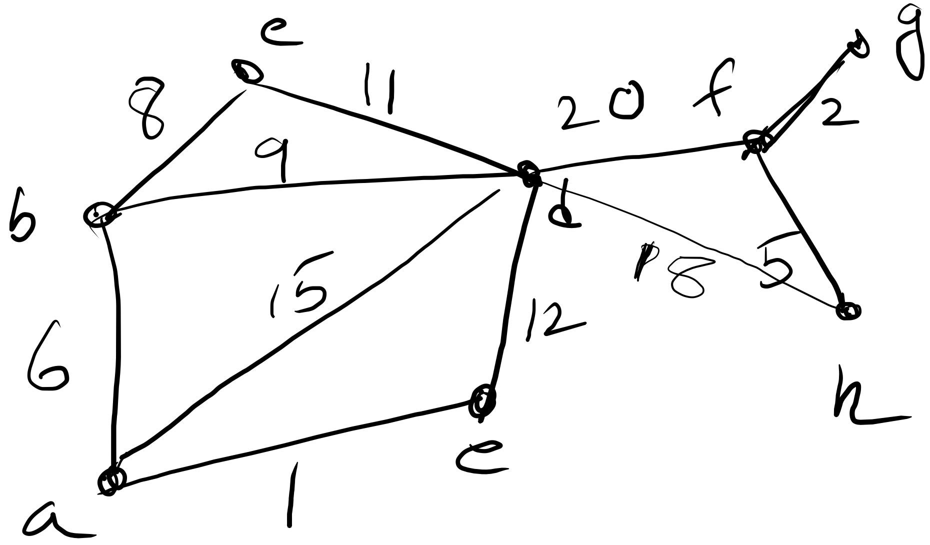


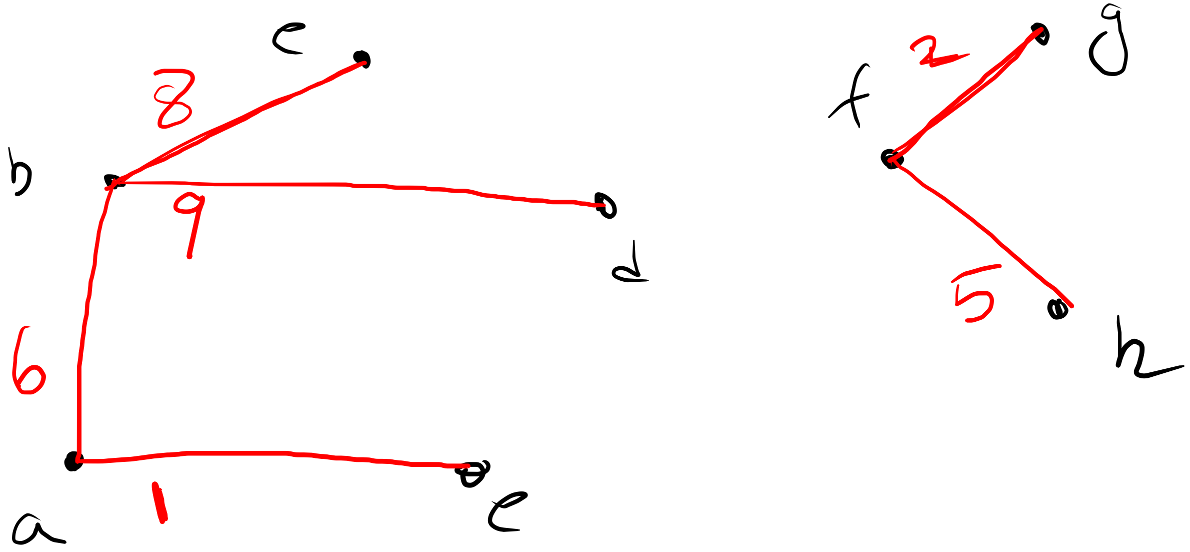


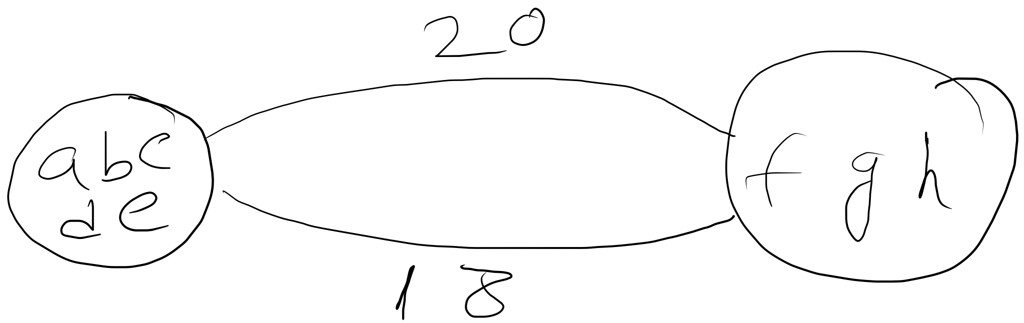
Boruvka's algorithm

Boruvka-MST(G, w)

- T is empty
  (assume that each vertex is a connected component)
- for each vertex $v \in V$
  - add the edge in T whose weight is minimum over all edges incident on v.
- G' be the graph generated by contracted all the edges of T
- T' be the MST computing recursively on G'
- return the set $T \cup T'$







$T' = (d, h)$ of weight 18

Reverse delete

Reverse-delete-MST(G, w)

- T is E
- sort the edges in E in non-increasing order by their weights.
- while E is not-empty.
    - choose e with largest weight
    - if removing e does not disconnect T
      remove e from T

return the set T.