Kruskal's algorithm : Efficient implementation

Kruskal-MST (G, w)

- T is empty
- Sort the edges based on non-decreasing weights.
- each vertex v is placed in a set $\leftarrow$ make set
- While E is not empty

  choose $e = (u, v) \in E$

  If u and v belongs to two different sets $\leftarrow$ $\mathrm{Find}(u) \neq \mathrm{Find}(v)$

  add e to T

  merge the sets containing u and v $\leftarrow$ Union (u, v)

- return the set T

Sorting $- O(|E| \log |E|)$

$|V| \leftarrow$ makeset operations.

$2|E| \leftarrow$ Find(u) "

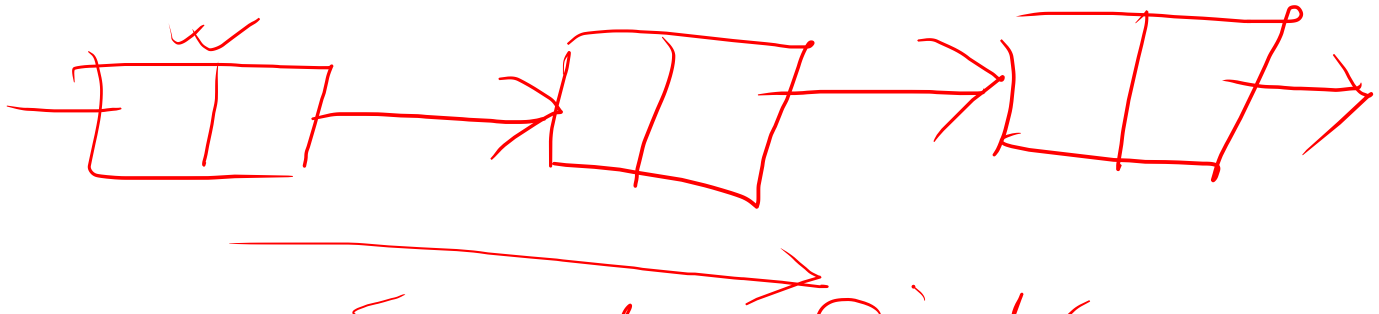
$|V|-1 \leftarrow$ Union "

Total running time $O(|E| \log |E|) + \underline{|E| \log |V|}$

If we implement disjoint set operations by union by rank.

then makeset $O(1)$ union $- O(\log |V|)$

Find(u) $- O(\log |V|)$



Search Find(u)

merge ,

Prim's algorithm

$\pi(v) \leftarrow$ minimum cost between v and s

Pred $[v] \leftarrow$ vertex just before v

Prim-MST (G, w) — T

- T is empty — $\theta(1)$
- $X = \{s\}$ — $\theta(1)$
- for each vertex $v \neq s$ — $O(|V|)$
    - $\pi[v] \leftarrow \infty$, Pred$[v] \leftarrow$ Null — $\theta(1)$
- $\pi[s] \leftarrow 0$ — $\theta(1)$
- create an empty priority queue Q — $\theta(|V|)$
- for each vertex $v \in V$ — $\theta(|V|)$
    - Insert $(Q, v, \pi[v])$ — T_{insert}
- while Q is not empty
    - $u \leftarrow$ Extract-min (Q)
    - for each edge $v \in Adj[u]$
        - if $v \in Q$ and $w(u,v) < \pi[v]$
            - Decrease-key $(Q, v, \pi[v])$
            - Pred $[v] \leftarrow u$

$O(|V|)$ times

$O(|E|)$ times

$T = O(|V|)\, T_{insert} +$

$O(|V|)\, T_{extractmin} +$

$O(|E|)\, T_{Decrease\text{-}key}$

Priority queue	$T_{extract\text{-}min}$	$T_{decrease\text{-}key}$	Total
Array	$O(\lvert V \rvert)$	$O(1)$	$O(\lvert V \rvert^2)$
Binary heap	$O(\lg \lvert V \rvert)$	$O(\lg \lvert V \rvert)$	$O(\lvert E \rvert \lg \lvert V \rvert)$
Fibonacci heap	$O(\lg \lvert V \rvert)$ amortized	$O(1)$ Amortized	$O(\lvert E \rvert \lg \lvert V \rvert)$ Amortized

Shortest path problem

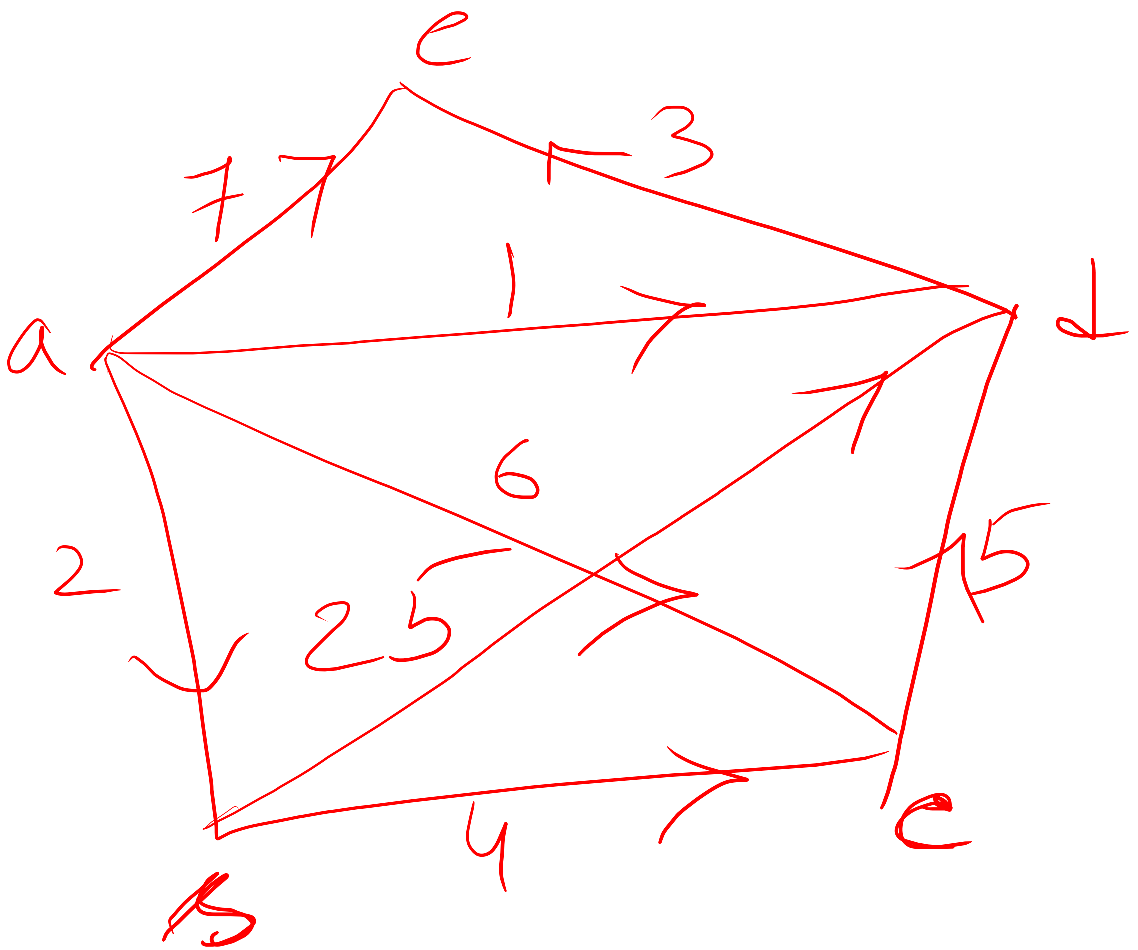
Given a (directed or undirected) graph $G(V, E)$ with edge costs $w : E \to \mathbb{R}^+$

<u>output</u>: i) Given $s, t \in V$ find shortest path from s to t

ii) Given $s \in V$ find shortest path from s to all other vertices

iii) Find shortest paths from all pairs of vertices.

single source shortest path



$d = t$

$s \to c \to d\,(=t)$ cost 9

$s \to d\,(=t)$ cost 25