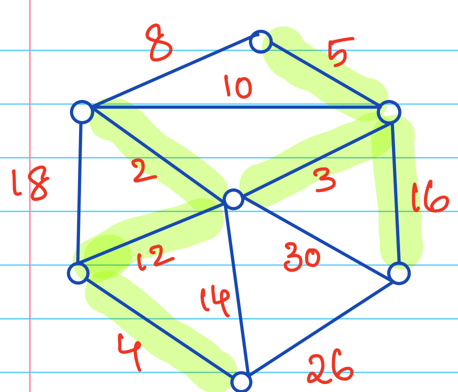
Minimum Spanning Trees

$G(V, E)$ - weighted undirected graph $w: E \to \mathbb{R}$

Tree $T(V, E')$ $E' \subseteq E$

$wt(T) = \sum_{e \in E'} wt(e)$

find the spanning tree T with smallest $wt(T)$



Theorem: If the edge weights are distinct, then G has a unique MST

Proof: Suppose not. Let T and T' be two MSTs. Since $T \neq T'$ $\exists\, e \in T \setminus T'$ & $e' \in T' \setminus T$

Assume, w.l.o.g $\underset{(\text{min } wt)}{wt(e)} < \underset{(\text{min } wt)}{wt(e')}$

Consider the graph $T' \cup \{e\} \to$ contains exactly one cycle (passing through e

Let e'' be the min-wt edge in C that is not in T

$$wt(e'') \geq wt(e') > wt(e)$$

Let $T'' = T' + \{e\} - \{e''\}$

$$wt(T'') = wt(T') + wt(e) - wt(e'')$$

$$wt(e) - wt(e'') < 0 \implies wt(T'') < wt(T')$$

This is not possible since T' is an MST

The only other possibility is $wt(e) = wt(e'')$

↳ again a contradiction

Assumption: Given graph G has unique edge weights (& this is w.l.o.g)

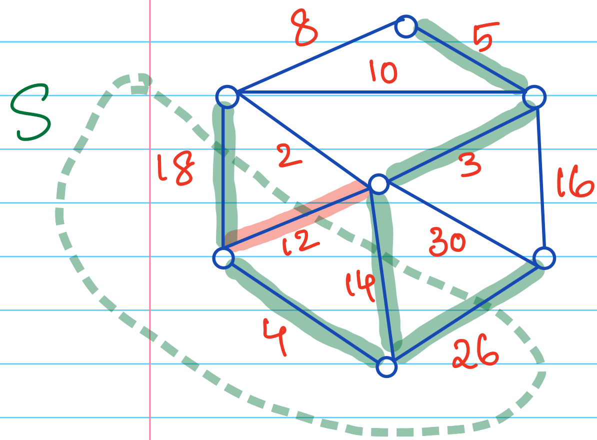
Theorem (Greedy choice):

Let $G(V, E, w)$ be a weighted undirected graph with distinct edge weights

Let $S \subseteq V$ be any subset, and let e be the min wt. edge with exactly one end point in S. Then e is part of the MST

S' is fixed, & e is the min-wt edge

Proof: Let T be the MST, and suppose that $e \notin T$



Let $e = (u, v)$

$\exists$ path from u to v in T

Let e' be any edge with exactly one end-point in S in this path $(wt(e') > wt(e))$

$T' = T - \{e'\}$: Spanning forest with exactly two components

$T'' = T' + \{e\}$: Spanning tree of G

$wt(T'') = wt(T) - wt(e') + wt(e)$

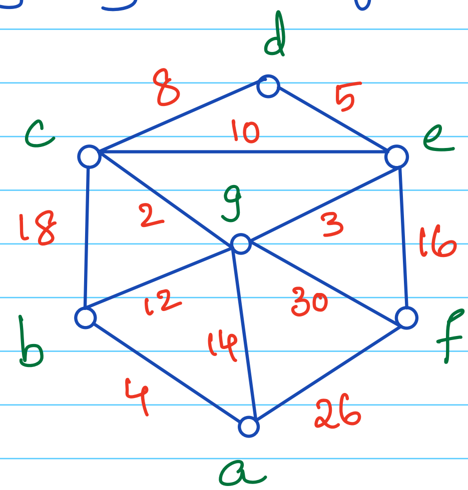
$< wt(T) \longrightarrow$ contradiction

Three instantiations of this greedy choice

1. Borůvka's algorithm
2. Prim's algorithm
3. Kruskal's algorithm

Borůvka's algorithm

At every step choose the min-wt edge going out from every component



- Intially each vertex in a Singleton component

Step 1: (a, b), (c, g), (f, e), (e, g), (e, d)

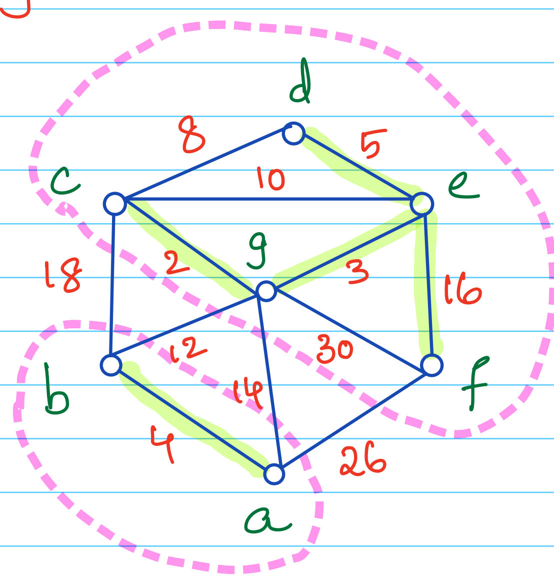
(a, b): chosen by a & b

(c, g): chosen by c & g

(f, e): chosen by f

(e, g): chosen by e

(e, d): chosen by d

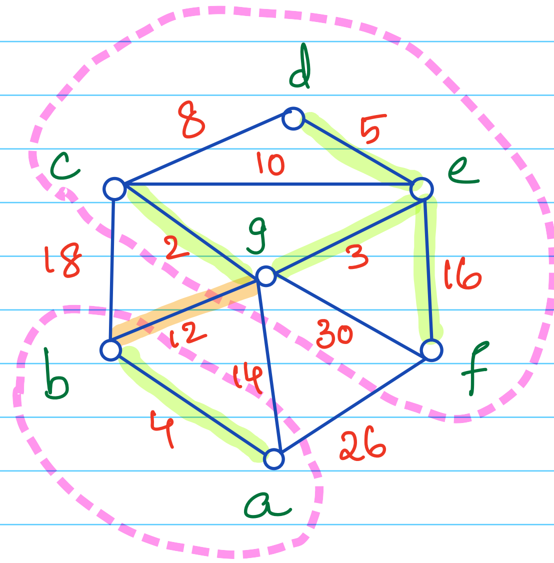


Step 2: (b, g)

↳ chosen by the component containing (a, b) & {c, g, f, e, d}

Final MST:

wt(MST) = 42



Implementation details

* At every step, ∃ spanning forest of G.

log n steps

- Run a DFS to label the components

$O(|V|+|E|)$

- For each edge check if it crosses a forest, & find the minimum

Total running time: $O((|V|+|E|)\log|V|)$