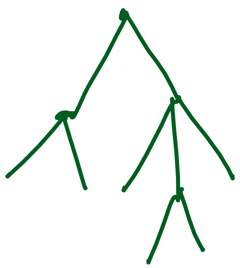
Minimum Spanning Trees (MST)

Def: A tree is an undirected graph which is connected and acyclic.

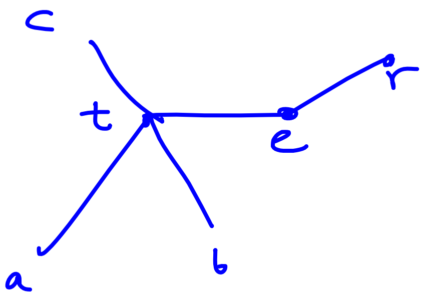


acyclic and connected graph

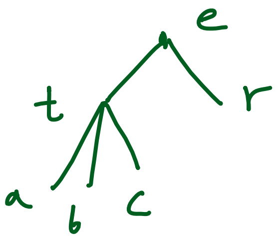
if we forget root and ordering among children of a node, it is a tree in the sense of definition above.

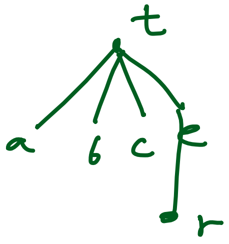
Rooted ordered trees

Conversely, if there is any connected acyclic graph



then we can choose any node as its root and draw it accordingly





Two rooted version of the tree given in blue above.

Two notions of tree define the same concept.

Spanning Tree of a graph

$G = (V, E)$

$T = (V, R)$ where $R \subseteq E$

s.t. T is a tree.

Weighted graphs $(V, E, w: E \to \mathbb{R})$

$$W(T) = \sum_{e \in R} w(e)$$

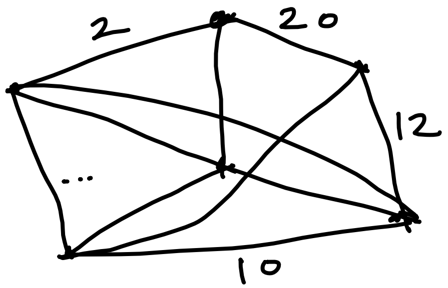
(sum of weights of edges in T)

MST: spanning tree (of G) with minimum weight.

Application (Example)

Villages : vertices

distance between v_1 v_2 is $w(v_1 v_2)$



Want to construct roads s.t. all villages are connected by these roads but want to minimize the total length of roads constructed

Solution is MST of the given graph.

Algorithms for finding MST

① Kruskal's Algo ② Prim's Algo.

Kruskal's Algorithm

Input $G = (V, E, w: E \to R)$

At any stage in the algorithm, there is a forest (V, F_i) $F_i \subseteq E$.

Initially, (V, F_0) $F_0 = \phi$ [so each node in G is a separate connected component]

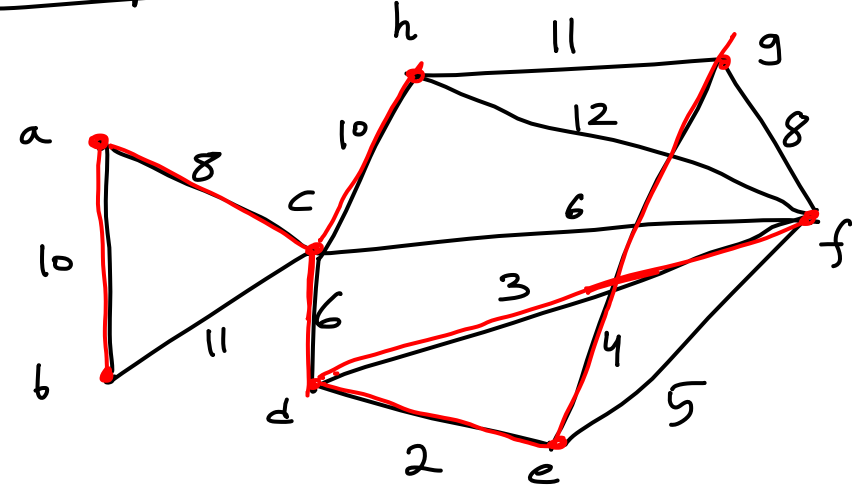
Keep growing the forest by adding edges to it using the following rule.

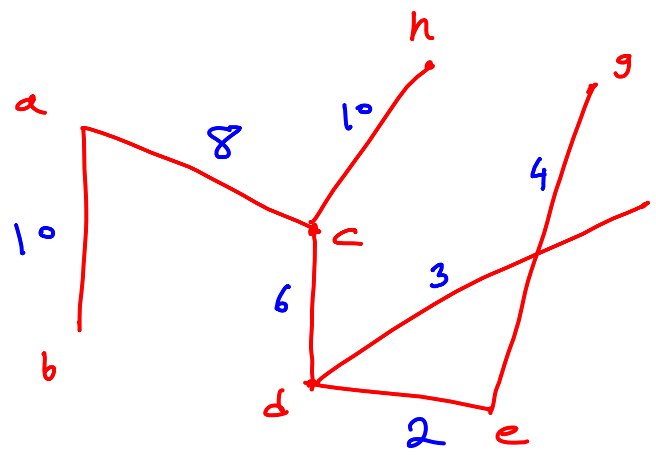
Add minimum weight edge (not already in F_i) if it is between two distinct components of F_i to get F_{i+1}

$n-1$ iterations suffice. F_{n-1} is the tree output by algorithm.

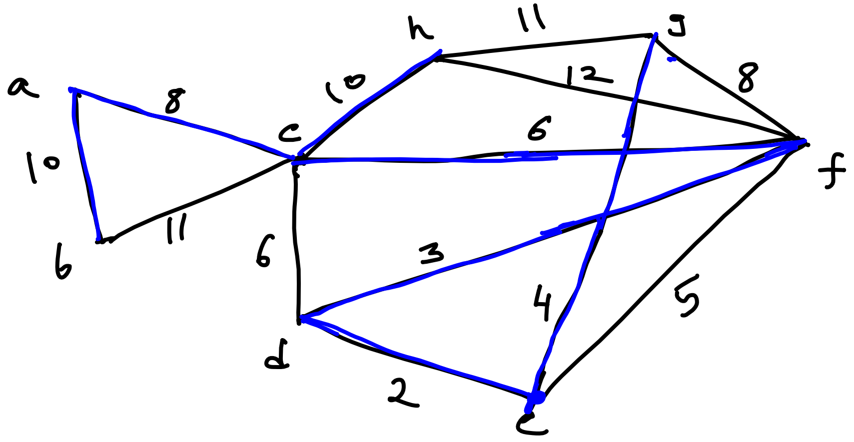
Kruskal's Algo

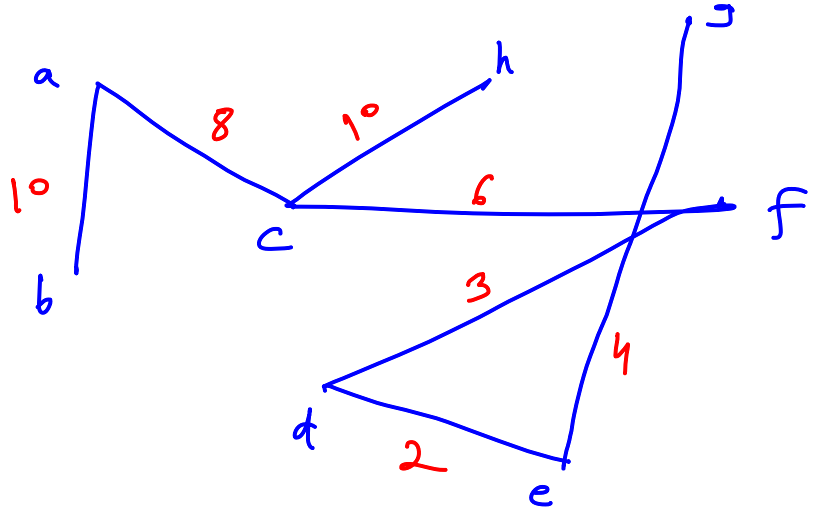
Example





43





43

Prim's Algorithm

Input $G = (V, E, w: E \to R)$

At any stage i, in the algo there is a tree T_i

Initially T_0 = some vertex of G.

T_{i+1} is constructed from T_i by adding a new vertex and a edge to T_i using the following rule.

Choose the edge with minimum weight that is between T_i an $V - T_i$. Add it to T_i to get T_{i+1}.

(Easy to see that T_{i+1} is connected, acyclic hence a tree)

output T_{n-1} $\quad (n = |V|)$

Correctness

We will prove correctness of a generic algorithm to find MST, and then see Kruskal and Prim's algorithms special case of this generic algorithm.

An edge (u,v) is safe for A if there is a MST containing $A \cup \{(u,v)\}$.

Input (G) // $G = (V, E, w)$

$A = \phi$

for $i = 1$ to $|V| - 1$ do

Let $(u,v) \notin A$ be a safe edge for A.

$A = A \cup \{(u,v)\}$

output A

Input graph G is assumed to be Connected.

(Every connected graph has a spanning tree)

Invariant: There is a MST of G containing A.

Invariant, which holds for final A $\Rightarrow$ output is a MST

[Final A has $|V|-1$ edges so it can't be properly extended to get a tree]

To find a safe edge. we need some definitions first.

Defn: A cut of $G = (V, E, w)$ is a partition of V into two parts $(S, V-S)$.

Defn: An edge (x,y) of G crosses a cut $(S, V-S)$ if $x \in S$ and $y \in V-S$.

Defn: A set 'A' of edges respects a cut $(S, V-S)$ if no edge in 'A' crosses $(S, V-S)$.

Defn: Let A be a set of edges of G which respects cut $(S, V-S)$. An edge (u, v) is light if it is the minimum weight edge which crosses the cut $(S, V-S)$.

Theorem (cut theorem)

Let $G = (V, E, w)$ be a weighted graph. Let $A \subseteq E$ respect cut $(S, V-S)$.
Let (u,v) be a light edge for cut $(S, V-S)$.
If there is a MST containing A then there is also a MST (of G) containing $A \cup \{(u,v)\}$.

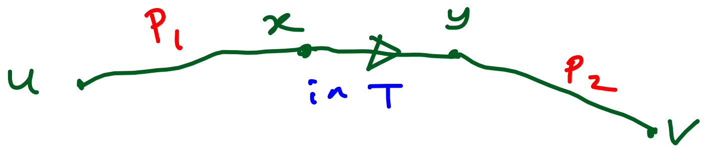
Proof: Let T be a MST containing A. (we know that one exists)
If $(u,v) \in T$ then the result already holds.
If $(u,v) \notin T$, then consider a path in T from u to v. This path must have an edge say (x,y) which crosses $(S, V-S)$, because u, v are on different side of the cut $(S, V-S)$.
By our assumption, $w(x,y) \geq w(u,v)$ [$\because$ (u,v) is a light edge]

Consider set of edges: $T' = (T - \{(x,y)\}) \cup \{(u,v)\}$

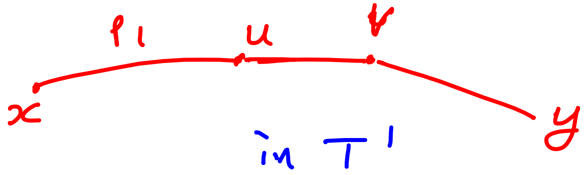
$$w(T') \leq w(T)$$

T' is a tree:

T' is connected: It suffices to show that there is a path from x to y in T'.



Simple path (edge (x,y) does not occur on P_1 or P_2)



$|T'| = |T|$ (we removed one edge and added another edge)

$\Rightarrow$ $|T'| = |V| - 1$ and T' is connected

$\Rightarrow$ T' is a tree.

$\Rightarrow$ $w(T') = w(T)$ ($\because$ T is a MST)

$T' \supseteq A \cup \{(u,v)\}$ T' is a MST $\square$

Correctness of kruskal's algo using cut theorem

Initially $A = \phi$ Invariant holds

F_i adding a min weight edge between two distinct components.

Let this edge be (a,b) $a \in C_r$, $b \in C_s$

Take $(C_r, V - C_r)$ as a cut.

(a,b) is a light edge for this cut.

F_i respects $(C_r, V - C_r)$

By cut theorem (a, b) is safe.

$\square$

Exercise: Prove correctness of Prim's algorithm using cut theorem.

(Easy).

Kruskal and Prim's algorithms are examples of greedy algorithms.