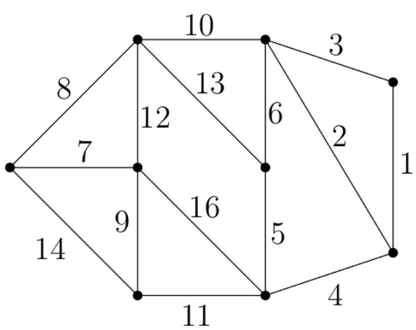
Lecture 31 Minimum Spanning Trees
Copyright Michael Monagan
December 2, 2020 11:22 AM

Lecture 31 Weighted Graphs and Minimum Spanning Trees

Grimaldi 13.2

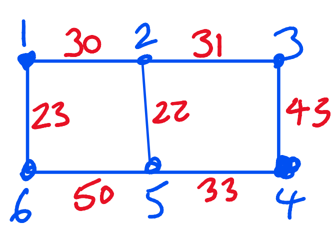


Definition (Weighted Graph)

A **weighted graph** $G = (V, E)$ is a multigraph together with a function $w : E \to \mathbb{R}^+$ is called an **edge-weighting**.

weights > 0.

Examples



Vertices :	cities	junctions	servers
Edges :	roads	pipes	fibre opt. cables
Weights :	distances	capacities	bandwidth.

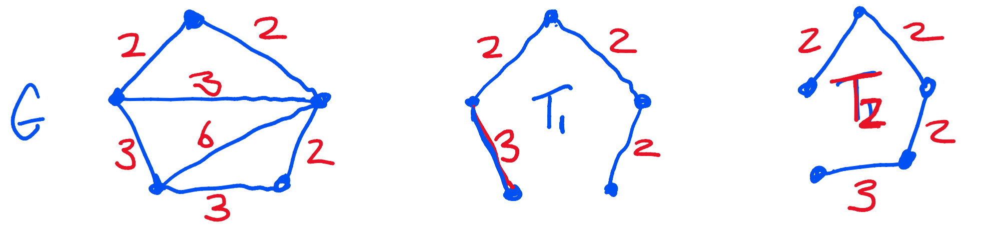
Definition (Minimum Spanning Tree)

Let $G = (V, E)$ be a connected multigraph with edge-weighting w.
For any subgraph $H = (V', E')$ of G, the **weight** of H is

$$w(H) = \sum_{e \in E'} w(e).$$

A **minimum spanning tree** is a spanning tree of G of minimum weight.

Example.



$|V|=5$
$|E|=4$

$w(G) = 3\times2+3\times3+6$
$= 6+9+6 = 21$

$w(T_1)=9$. A M.S.T.

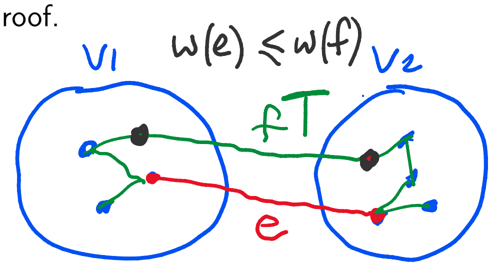
$w(T_2)=9$

Lemma (property of minimum spanning trees)

Let $G = (V, E)$ be a weighted connected graph. Let V_1 and V_2 be a partition of V. Amongst the edges in G with one vertex in V_1 and the other in V_2 let e one of minimum weight. There is a minimum spanning tree in G with e as one of it's edges.

Idea: chose an edge e of least weight.

Proof.



Let T be a M.S.T. in G. If T does not contain e then adding e to T must create a cycle C. There must be an edge f on C with one vertex in V_1 and the other in V_2.
Let $S = T \cup \{e\} - \{f\}$. S is a spanning tree with $w(S) \leq w(T)$ because $w(e) \leq w(f)$.
Since T is a M.S.T. then S must be too.

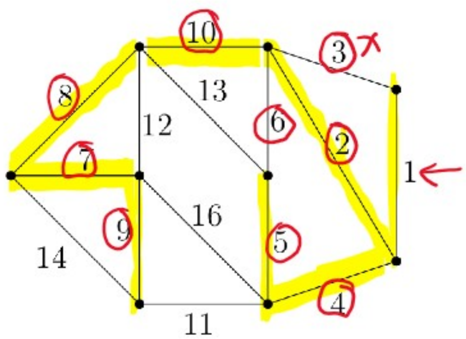
Kruskal's algorithm to compute a minimum spanning tree

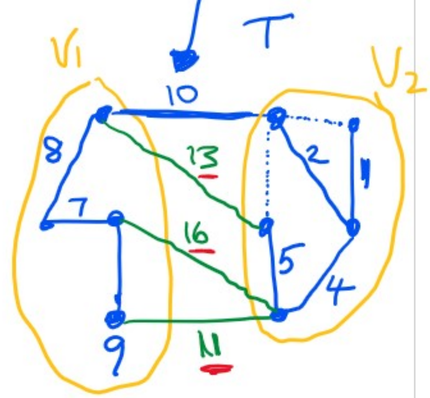
Input: a connected multigraph $G = (V, E)$ with an edge-weighting w.
Output: a mimimal spanning tree of G.

1. Set $E' = \phi$. $T = (V, E')$
2. Sort the edges in E from least weight to highest weight. ← mergesort algorithm.
3. While T is not connected do while $|E'| < |V| - 1$ do
   Let e be the next heaviest edge in E. has least weight among the choices at this step
   If $(V, E' \cup \{e\})$ does not have a cycle set $E' = E' \cup \{e\}$.
4. Return the tree (V, E').

$|V| = |E| + 1$
$|E| = |V| - 1$

Example





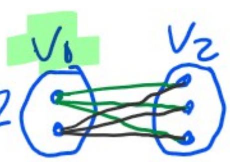
Additional Space.

P_n path on n vertices e.g. $P_3 =$ •—•—• length 2
$n \geq 1$ $P_1 =$ • length 0

C_n cycle on n vertices e.g. $C_3 =$ $C_1 =$ $C_2 =$
$n \geq 1$

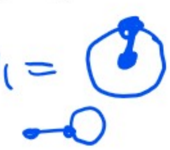
K_n complete graph on n vertices $K_4 =$ $K_1 =$ •
$n \geq 1$

$K_{m,n}$ complete bipartite graph on $m+n$ vertices
$m \geq 1$ $n \geq 1$. $K_{1,1} =$ •—•



W_n wheel graph on n spokes e.g. $W_3 =$
$n \geq 1$ W_1

$W_1 =$

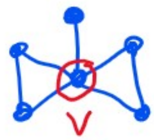


What are the smallest values for m, n ?

Additional Space.

A.P.s

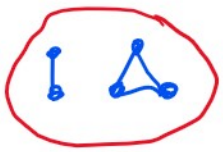
Cut Vertex.

G = 

G − v 

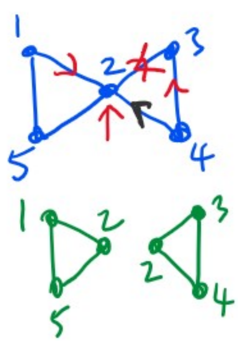
Def:
A vertex v in a graph G is an A.P.
if G − v has more connected components than G.

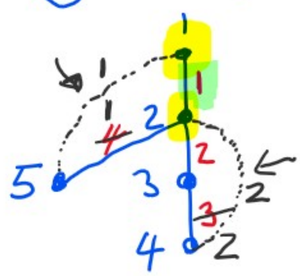
Biconnected. Def: Let G be a graph. A subgraph of G is biconnected if it has no APs. and is connected.

G =  biconnected?

No APs

How do we find APs and BCs of a graph.





① Find a DFS spanning tree.

② Number the back edges.

③ Identify the APs and BCs.

APs? 2

BCs?