Lec32 Minimum Spanning Trees Michael Monagan

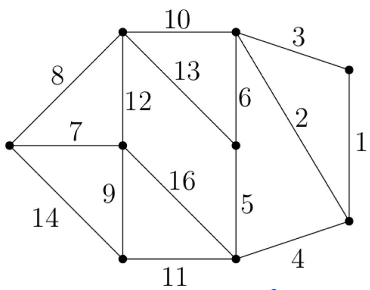
April 14, 2021 11:10 AM

Lecture 32: Weighted Graphs and Minimum Spanning Trees

Grimaldi 13.2

Assignment #8 due Friday @ 11pm
Covers Lectures 29–32 (today).

Please do the course evaluation.



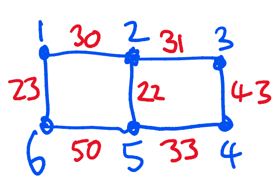
A Weighted Graph.

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**.

$> 0.$

Examples



Vertices :	cities	junctions	servers
Edges :	roads	pipes	cables
Weights :	distances	capacity	bandwidth.

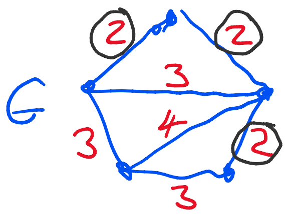
Definition (Minimum Spanning Tree)

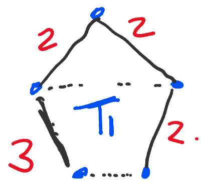
Let $G = (V, E)$ be a connected multigraph with edge-weighting w.
For any subgraph $H = (V', E')$ of G, the **weight** of H is

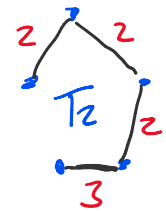
$$w(H) = \sum_{e \in E'} w(e).$$

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

Example.







$W(G) = 3\times 2 + 3\times 3 + 4$
$= 6+9+4 = 19.$

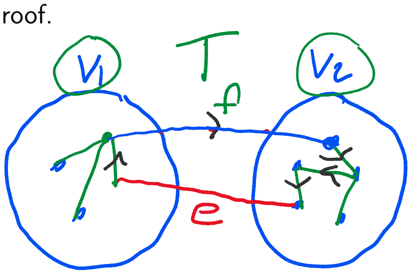
A M.S.T.
$W(T_1) = 9.$

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.

Proof.



Let T be a M.S.T. of G.
If T does not have 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 one in V_2.
Let $S = T - \{f\} \cup \{e\}$. 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

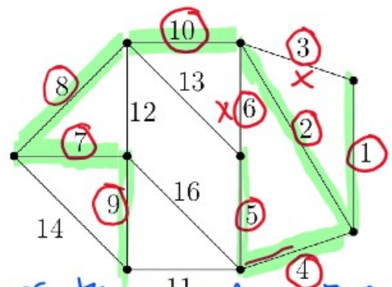
Input: a connected multigraph $G = (V, E)$ with an edge-weighting w.
Output: a mimimal spanning tree of G.

1. Set $E' = \phi$. for the M.S.T.
2. Sort the edges in E from least weight to highest weight.
3. While $\boxed{(V, E') \text{ is not connected}}$ do   while $|E'| < |V| - 1$ do
   Let e be the next heaviest edge in E.
   If $(V, E' \cup \{e\})$ does not have a cycle set $E' = E' \cup \{e\}$.
4. Return the tree (V, E').

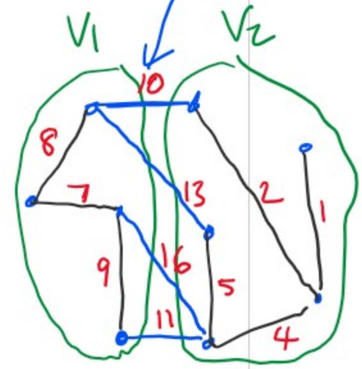
Stop when $\boxed{\begin{array}{l} |V| = |E'| + 1 \\ |E'| = |V| - 1 \end{array}}$

this edge has least weight among those edges connecting V_1 & V_2.

Example



Stop as the green tree is connected.



Additional Space. Draw

P_4  length 3 edges

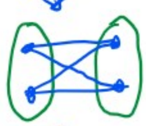
P_n is a path graph on n vertices
$n \geq 1$ $P_1 = \bullet$

C_4 

C_n is a cycle on n vertices
? multigraphs $n \geq 1$
$C_1 =$  $C_2 =$  $C_3 =$ 

K_4 

K_n the complete graph on n vertices
$n \geq 1$. $K_1 = \bullet$

$K_{2,2}$ 

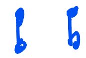
$K_{m,n}$ the complete bipartite graph
$m \geq 1, n \geq 1$ $K_{1,1}$ 

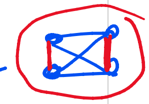
W_4 

W_n the wheel graph on n spokes.
$n \geq 1$. W_1 

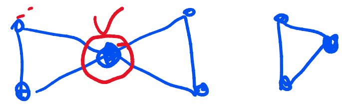
Question: What is the smallest value(s) for n (and m).

Additional Space.

$\overline{K_{2,2}}$ 

$K_{2,2} =$ 

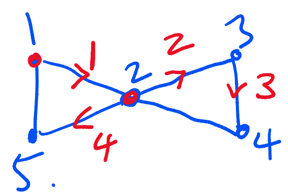
A.P.s (cut vertexes)

G 

Def A vertex v in a graph G is an articulation point if $G-v$, $G-v$ has more connected components than G.

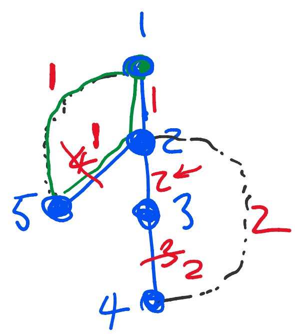
Def. Let G be a graph. A subgraph of G is biconnected if G has no A.P.s and G is connected.

How do we find the A.P.s and B.C.s. (maximal biconnected subgraphs).



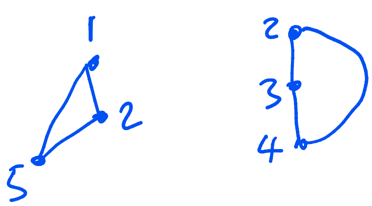
Bowtie Graph.

① Find a DFS Spanning Tree



APs. 2.

BCs.



② Include back edges. and renumber the edges and back edges.

③ Read off the APs & BCS