Module-7

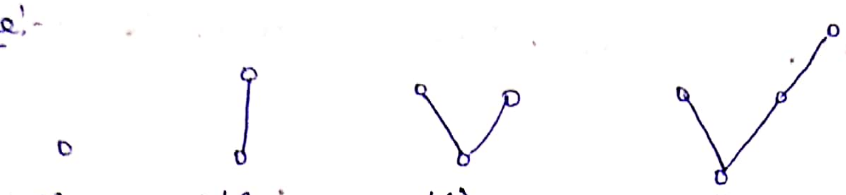
– R. Udhayakumar

Trees – Properties of trees

Tree

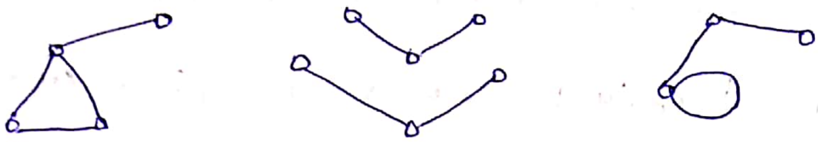
A tree is a connected graph without any cycles.

Example:-



(a) (b) (c) (d)

(a, b, c & d) are trees.



(e) (f) (g)

e, f, g are not trees.

Some properties of trees

Property-1 An undirected graph is a tree, if and only if, there is a unique simple path between every pair of vertices.

Proof:- Let the undirected graph T be a tree.

($\Rightarrow$) Then by the defn of tree, T is connected.

Hence, there is a simple path b/w any pair of vertices, say v_i and v_j.

If possible, let there be two paths b/w v_i and v_j – one from v_i to v_j and the other from v_j to v_i.

Combine (union) of these paths would contain a cycle, which is a $\Rightarrow\Leftarrow$ since T is a tree.

$\therefore$ Hence the simple path v_i to v_j is unique.

$(\Leftarrow)$ Let a unique path exists b/w every pair of vertices of T.

Then T is connected.

If possible, let T contain a cycle. This means that there is a pair of vertices v_i and v_j b/w which two distinct paths exists, which is $\Rightarrow\Leftarrow$.

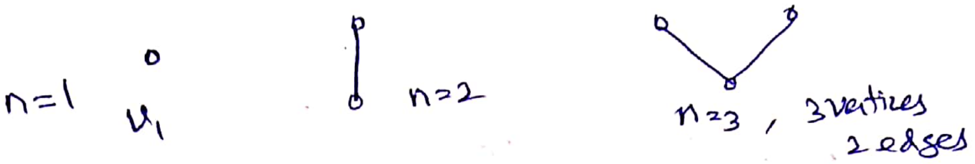
Hence T cannot have a circuit & so T is a tree.

Property II

(X) A tree with n vertices has (n-1) edges.

Proof:- Proof by Mathematical induction.

The property is true for $n=1, 2, 3$ given by example.



Assume the property is true for all trees with less than n vertices.

Let us consider a tree T with n vertices.

Let e_K be the edge connecting the vertices v_i and v_j.

Then by property (I), e_K is the only path b/w v_i & v_j.

If we delete the edge e_K from T, T becomes disconnected. and $(T-e_K)$ consists of exactly two components say T_1 and T_2 which are connected.

Since T did not contain any cycle, T_1 & T_2 also will not have cycles.

Hence, both T_1 and T_2 are trees, each having less than n vertices say r and $n-r$ respectively.

$\therefore$ By induction T_1 has $r-1$ edges & T_2 have $n-r-1$ edges.

$\therefore$ T has $(r-1)+(n-r-1)+1 = n-1$ edges

Thus a tree with n vertices has $(n-1)$ edges.

Property III

Any connected graph with n vertices and $(n-1)$ edges is a tree.

Property. IV

Any ~~cycles~~ circuitless graph with n vertices & $(n-1)$ edges is a tree.

Distance and centres in a Tree

Distance of the graph

In a connected graph G, the distance $d(v_i, v_j)$ between two of its vertices v_i and v_j is the length of the shortest path b/w them.

Eccentricity

Let G be any graph. Consider any vertex $u \in G$ and its distances from all other vertices. The maximum of these distances is called the eccentricity of the vertex u and it is denoted by $e(u)$.

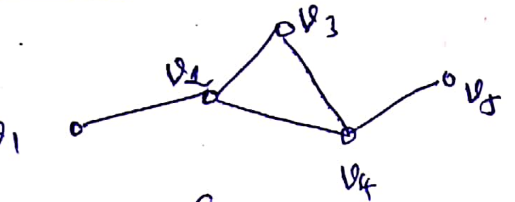
Radius

Consider the eccentricities of all the vertices of the Graph G. The minimum of these eccentricities is called the radius of the graph and it is denoted by $r(G)$.

Central point

If a vertex u of a graph G is such that its eccentricity is equal to the radius of the graph ($e(u) = r(G)$) then the vertex u is called a central point. The set of all central points of the graph is called the centre of the graph and it is denoted by $C(G)$.

Example:-



G

Distance of v_1	Distance of v_2
$d(v_1, v_2) = 1$	$d(v_2, v_1) = 1$
$d(v_1, v_3) = 2$	$d(v_2, v_3) = 1$
$d(v_1, v_4) = 2$	$d(v_2, v_4) = 1$
$d(v_1, v_5) = 3$	$d(v_2, v_5) = 2$
$e(v_1) = 3$	$e(v_2) = 2$

III'ly find all other distances of vertices.

and $e(v_3) = 2$, $e(v_4) = 2$ & $e(v_5) = 3$.

The minimum of all eccentricities is 2.

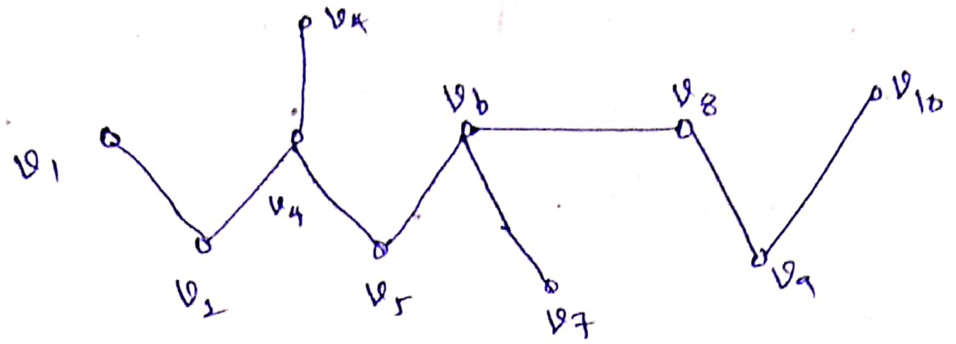
$\therefore$ The radius $r(G) = 2$.

We have find $e(v_2) = e(v_3) = e(v_4) = r(G) = 2$

$\therefore$ v_2, v_3 & v_4 are central points of G.

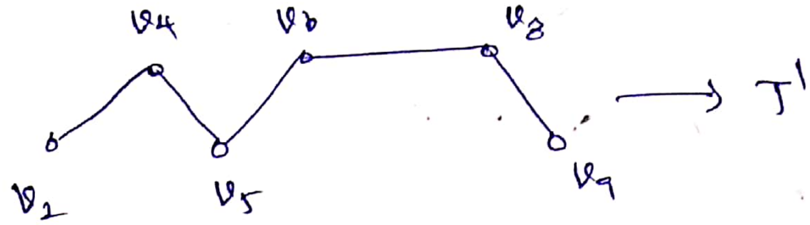
$\therefore$ $\{v_2, v_3, v_4\}$ is the centre of the graph.

(2) Find the centre of the following tree T.

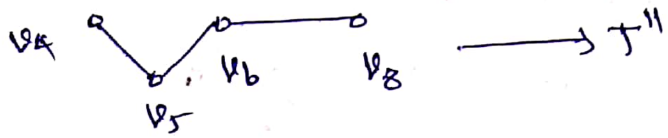


Soln:- Deleting the pendant vertices of T

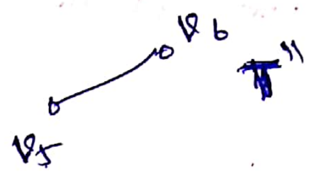
Step: 1



Step: 2 Deleting the pendant vertices of T'



Step: 3 Deleting the pendant vertices of T''.



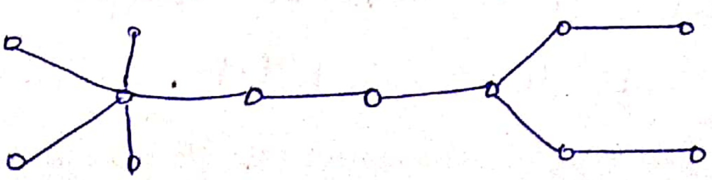
The given tree has a centre of two adjacent vertices

centre $C(T) = \{v_5, v_6\}$

Result:- Every tree has either one or two centres.

Example:-

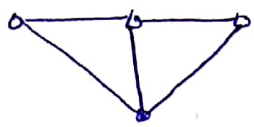
Find centre



Spanning Trees

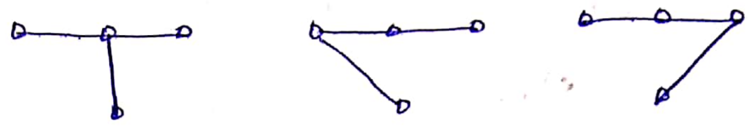
If the subgraph T of a connected graph G is a tree containing all the vertices of G, then T is called a spanning tree of G.

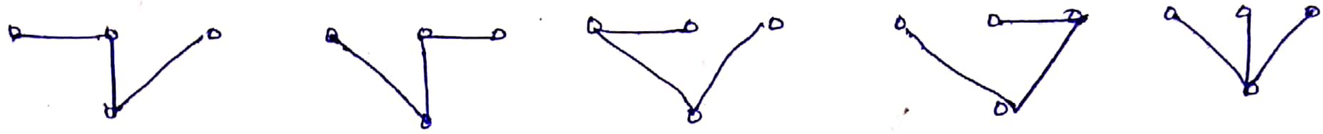
Consider the graph



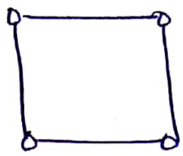
G

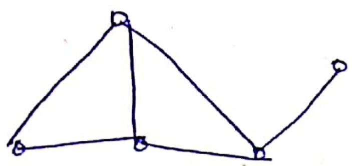
then





Since G has 5 edges, removal of 2 edges may result in spanning tree. this can be done in ${}^5C_2 = 10$ ways, but 2 of these 10 ways is gives disconnected graphs. All the possible spanning trees are shown in above.

(2) G  Find all the spanning trees of the graphs

(3) G 

Minimum Spanning tree

If G is connected weighted graph, the spanning tree of G with the smallest total weight (via the sum of the weights of its edges) is called the minimum spanning tree of G.

Fundamental circuits

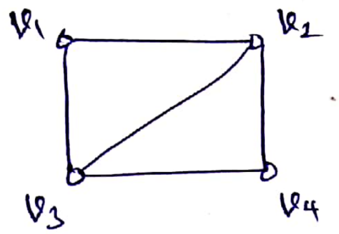
Branch:- Let G be a connected graph and T be a spanning tree of G. Every edge of the spanning tree T is called a branch of T.

Chord: An edge of the graph G which is not in the spanning tree T is called a chord of the tree T.

Fundamental circuit

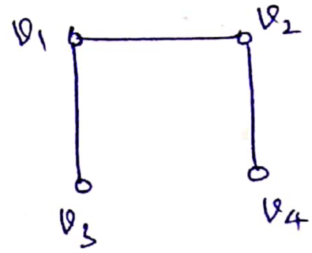
Let us consider a spanning tree T in a connected graph. Adding any one chord of T will create exactly one circuit. Such a circuit formed by adding a chord to a spanning tree, is called a fundamental circuit.

Example:- Find the fundamental circuit in the given graph G.



Soln:- The tree has branches

$\{v_1, v_2\}$ $\{v_1, v_3\}$ $\{v_2, v_4\}$



the remaing edges of G are $\{v_2, v_3\}$ & $\{v_3, v_4\}$.

Adding an edge $\{v_2, v_3\}$ to the tree then this creates a circuit $v_1 \to v_3 \to v_2 \to v_1$.

Illy adding $\{v_3, v_4\}$ we will get $v_1 \to v_3 \to v_4 \to v_2 \to v_1$

These two circuits are called fundamental circuits

Kruskal's Algorithm

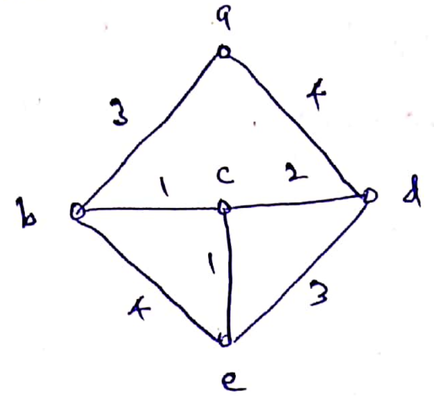
Step: 1 List all edges of the graph G in order of non-decreasing weight.

Step: 2 Select a smallest edge of G.

Step: 3 For each successive step select (from remaining edges of G) another smallest edge that makes no circuit with the previously selected edges.

Step: 4 If G has n vertices, stop after (n-1) edges have been chosen. Otherwise repeat step: 3.

Example: 1. Using Kruskal's algorithm, find a minimal spanning tree for the graph of the following graph.



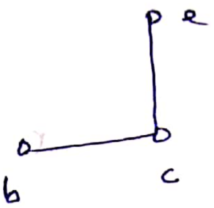
Soln:-

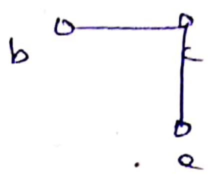
Edge	Weight
(b, c)	1
(c, e)	1
(c, d)	2
(a, b)	3
(e, d)	3
(a, d)	4
(b, e)	4

The steps for finding a minimal spanning tree are shown below:

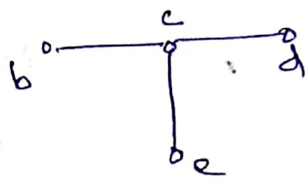
(i) choose the edge (b,c) as it has a minimal weight.

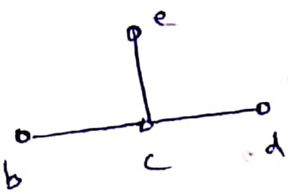
(ii) Add the next edge (c,e)



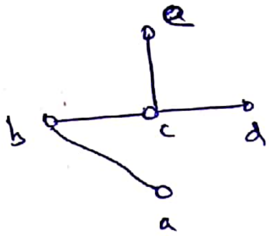


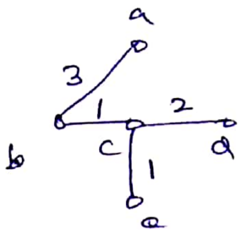
(iii) Add the edge (c,d)



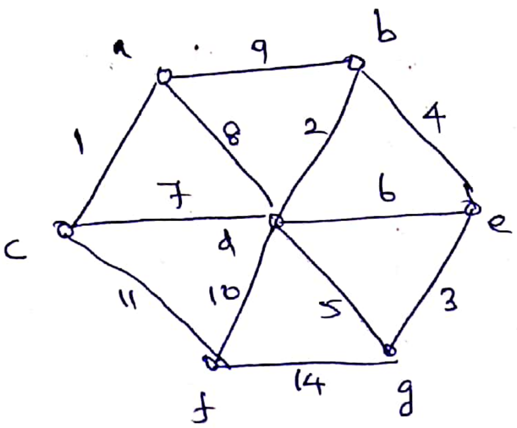


(iv) Add (a,b)





(1)



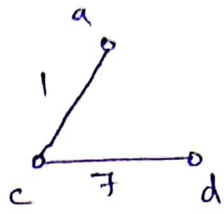
Soln:-

(a,c)	(b,d)	(e,g)	(b,e)	(d,g)	(d,e)	(d,c)	(a,d)	(a,b)
1	2	3	4	5	6	7	8	9

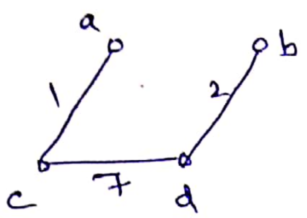
(d,f)	(c,f)	(f,g)
10	11	14

① Choose (a, c)

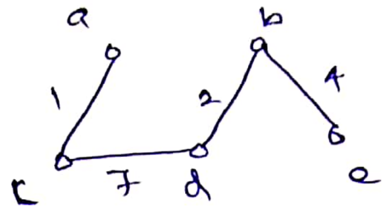
② Add next edge (c, d)



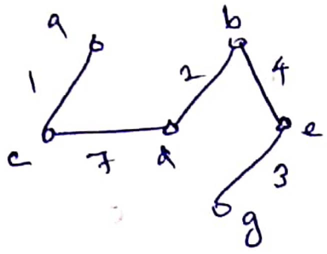
③ Add (d, b)



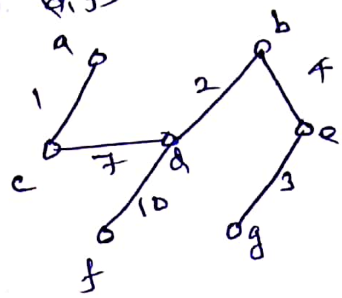
④ Add (b, e)



⑤ Add (e, g)



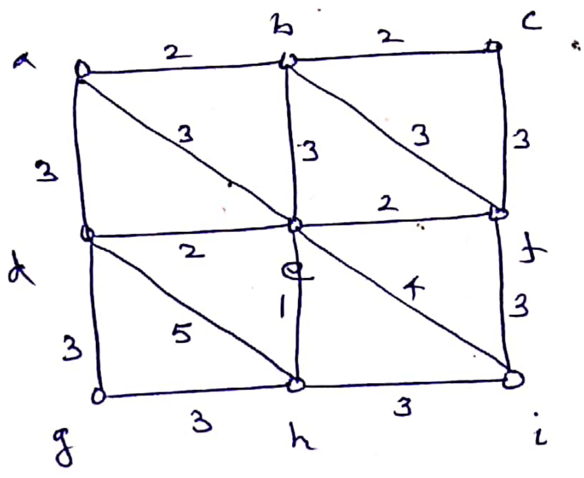
⑥ (d, f)



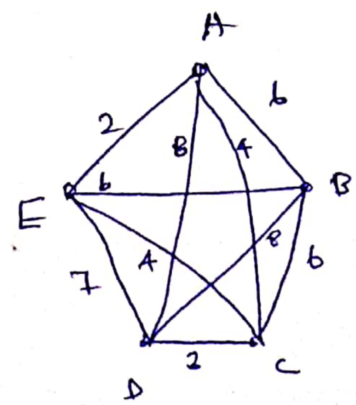
Since, vertices are 7 and we have 6 edges, we stop the algorithm and the minimum spanning tree is produced.

H.W.

①



②



Prim's Algorithm

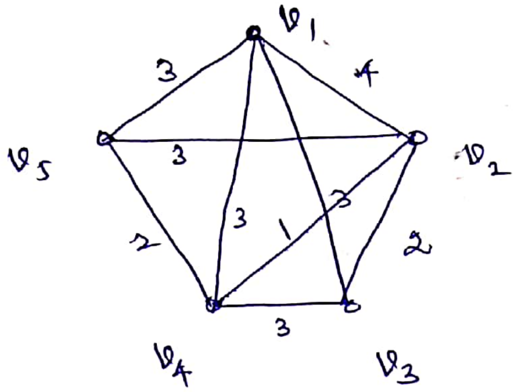
Step: 1 choose any vertex v_1 of G.

Step: 2 choose an edge $e_1 = v_1v_2$ of G s.t $v_2 \neq v_1$ and e_1 has smallest weight among the edges of G incident with v_1.

Step: 3 If edge $e_1, e_2, \dots e_i$ have been choosen involving end points $v_1, v_2, \dots, v_{i+1}$. choose an edge $e_{i+1} = v_jv_k$ with $v_j \in \{v_1 \dots v_{i+1}\}$ and $v_k \in \{v_1, \dots, v_{i+1}\}$. Such that e_{i+1} has smallest weight among the edges of G with precisely one end in $\{v_1, \dots, v_{i+1}\}$

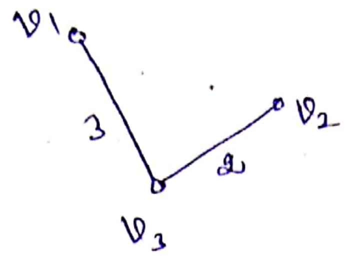
Step: 4 Stop after n-1 edges have been choosen. otherwise go to step: 3.

Example Find the minimal spanning tree of the weighted graph of the following graph using Prim's algorithm.

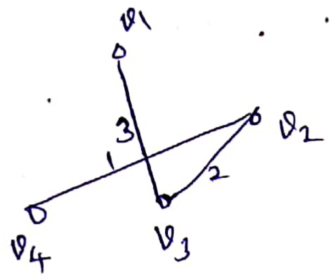


Soln:- 1. we choose the vertex v_1. Now edge with smallest weight incident on v_1 is (v_1, v_3). so we chose the edge.

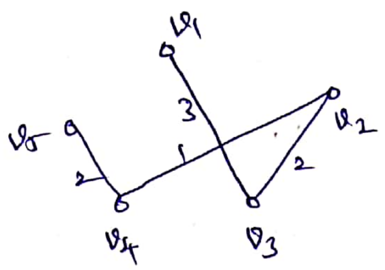
(2) we choose the edge $w(v_3, v_2) = 2$ since it is minimum.
we choose the edge (v_3, v_2).



(3) $w(v_2, v_4) = 1$ it is min.



(4) choose (v_4, v_5)



H.W

(1)