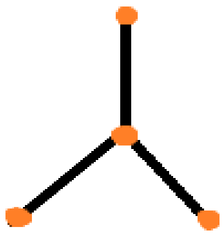
Tree

A **connected acyclic graph** is called a tree. In other words, a connected graph with no cycles is called a tree. It is usually denoted by T.

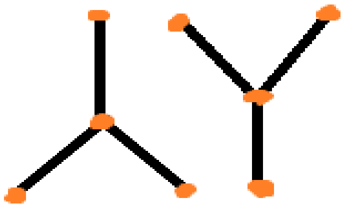
Forest

A **disconnected acyclic graph** is called a forest. In other words, a disjoint collection of trees is called a forest. Thus, a forest is nothing but a collection of trees.

Example:



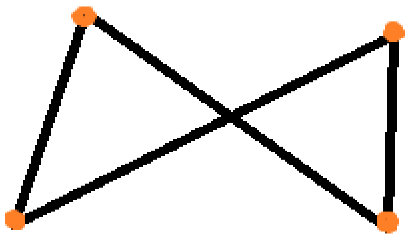
Tree



Forest

Example:

This is not a tree as it contains a cycle.



Result: In a tree (T), there is one and only one path between every pair of vertices in a tree.

Since tree (T) is a connected graph, there exist at least one path between every pair of vertices in a tree (T). Now, suppose between two vertices a and b of the tree (T) there exist two paths. The union of these two paths will contain a circuit and hence cannot be a tree. Hence the above statement is proved.

Result: If in a graph G there is one and only one path between every pair of vertices then graph G is a tree.

There is the existence of a path between every pair of vertices so the graph G is connected. A circuit in a graph implies that there is at least one pair of vertices a and b, such that there are two distinct paths between a and b. Since G has one and only one path between every pair of vertices. G cannot have any circuit. Hence graph G is a tree.

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

Let n be the number of vertices in a tree (T).
If n=1, then the number of edges=0.
If n=2 then the number of edges=1.
If n=3 then the number of edges=2.

Hence, the statement (or result) is true for n =1, 2, 3.

Let the statement be true for n=m. Now we want to prove that it is true for $n = m + 1$.

Let e be the edge connecting vertices, say v_i and v_j. Since G is a tree, then there exists only one path between vertices v_i and v_j.
Hence if we delete edge e it will disconnect the graph into two components G_1 and G_2, say.
These components have less than m+1 vertices and therefore by induction hypothesis has no circuit and hence each component G_1 and G_2 has m_1 and m_2 vertices (say), and $m_1 - 1$ and $m_2 - 1$ edges.

Hence the total edges $= (m_1 - 1) + (m_2 - 1) + 1$ [1 is added for the edge e, that was deleted]
$= (m_1 + m_2) - 1 = m + 1 - 1$ [as, $m_1 + m_2 = n = m + 1 =$ total vertices]
$= m$

Thus, for $n = m + 1$ vertices there are exactly m edges in a tree (T).
Hence, By the mathematical induction the graph exactly has $n - 1$ edges.

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

We know that the minimum number of edges required to make a graph of n vertices connected is $(n - 1)$ edges. We observe that removal of one edge from the graph G will make it disconnected.
Thus, a connected graph of n vertices and $(n - 1)$ edges cannot have a circuit.
Hence a graph G must be a tree.

Result: A graph with n vertices, $n - 1$ edges and no circuit is a connected graph.

Let, the graph G is disconnected. Then there exist at least two components G_1 and G_2, say.
Each of the component is circuit-less as G is circuit-less.
Now let us add one edge e between the vertices v_i, v_j, where v_i is a vertex of G_1 and v_j is a vertex of G_2.
Since there was no path between v_i, v_j in G, adding e will not create a circuit.
Now the number of edges in $G \cup e = n - 1 + 1 = n$.
Hence, $G \cup e$ is a circuit-less connected graph and hence a tree with n vertices and n edges.
But we know that a tree with n vertices must have exactly $n - 1$ edges.
Which is a contradiction and hence the result.

Result: A graph G is a tree if and only if it is minimally connected.

Let the graph G is minimally connected, i.e; removal of one edge make it disconnected.
Therefore, there is no circuit. Hence graph G is a tree.
Conversely, let the graph G is a tree i.e; there exists one and only one path between every pair of vertices.
We know that removal of one edge from the path makes the graph disconnected.
Hence graph G is minimally connected.

Leaf:

Let T be a tree. If $v \in V(T), d(v) = 1$, then v is called a ***leaf***.

Result: Every tree with at-least two vertices has at-least two leaves (i.e. two pendant vertices).

Let the number of vertices in a given tree T is $n(\geq 2)$.
Therefore, the number of edges in $T = n - 1$, using above results.
Then $\sum_{v \in V} d(v) = 2(n-1) = 2n - 2$.

This degree sum is to be distributed among n vertices.
As T is connected no vertex can have degree zero.
Each vertex contributes at-least one to the above sum.
Thus, there must be at least two vertices of degree one.
[otherwise, let all vertices have a minimum degree two; then $d(v) = 2n > 2n - 2$; a contradiction]

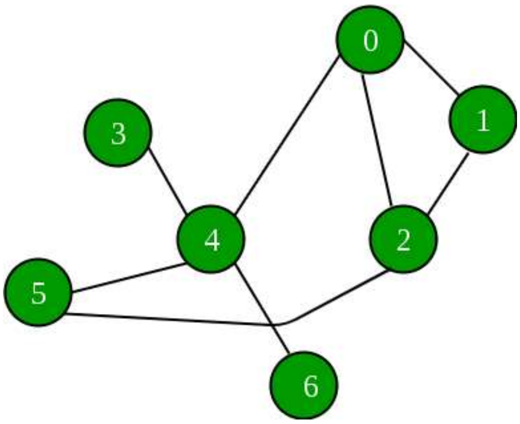
Hence every tree with at-least two vertices have at-least two pendant vertices or leaves.

Distance between two vertices:

The distance between two vertices in a connected graph G, denoted by $d(v_i, v_j)$, is the number of edges in a shortest or minimal path between v_i and v_j.

Note: There can exist more than one shortest path between two vertices.

Example:



In the graph on the left:

$d(1,5) = 2\ [1 \to 2 \to 5]$; $d(1,3) = 3\ [1 \to 0 \to 4 \to 3]$

Similarly, $d(4,3) = 1[4 \to 3]$; $d(4,2) = 2[4 \to 0 \to 2]$

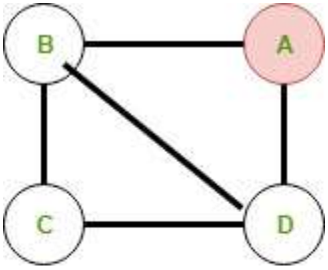
And, $d(6,6) = 0$; $d(6,5) = 2[6 \to 4 \to 5]$;
$d(6,1) = 3[6 \to 4 \to 0 \to 1]$

And so on…

Eccentricity of a vertex:

It is defined as the maximum distance of a vertex v from other vertices, denoted by $e(v)$.

Example:



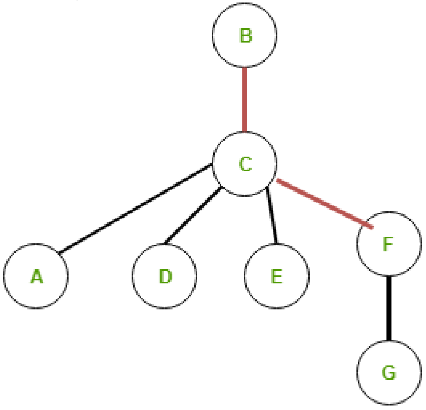
In the graph on the left,

$$e(A) = 2\ [\because d(a,c) = 2]; e(B) = 1\ ; e(C) = 2\ ; e(D) = 1$$

Centre of a graph:

A vertex is called the centre of a graph if it has the minimum eccentricity.

Example:



In the graph,

$e(B) = 3; e(C) = 2; e(A) = 3; e(D) = 3; e(E) = 3; e(F) = 2; e(G) = 3$

Hence, center of the graph is C & F

Note: This graph is also a tree

Result: Every tree has either one or two centres.

We will use one observation that the maximum distance $\max d(v, w)$ from a given vertex v to any other vertex w occurs only when w is a pendant vertex.

Now, let T be a tree with n vertices $(n \geq 2)$

T must have at least two pendant vertices.

Delete all pendant vertices from T, thus resulting in a graph T' which is still a tree.

Again, delete pendant vertices from T' so that resulting T'' is still a tree with same centers.

Note that all vertices that T had as centres will still remain centres in $T \to T' \to T'' \to \cdots$

Continue this process until the remaining tree has either one vertex or one edge.

So in the end, if one vertex is there, it implies that the tree T has one center.

If one edge is there then tree T has two centres (the end points of the edge).

Corollary:

From the argument above it is clear that if a tree has two centres, then they must be adjacent.

Result: If a forest G of order n has k components, its size is $n - k$.

Let, G be a forest of order n and k components as $G_i, i = 1, \dots k$.
Then each G_i is a tree of length say, $n_i, i = 1, . . , k$, with $n = \sum_{i=1}^{k} n_i$

Since G_i is a tree, then size of $G_i = n_i - 1$.
Hence, size of $G = \sum_{i=1}^{k}(n_i - 1) = n - k$

Result: An edge e of a graph G is a bridge (Cut edge) iff e does not lie on a cycle of G.

Proof by contradiction.
Suppose e is a bridge of G and on a cycle of G. By removing e, we know there is a path between two vertices of e. Which is a contradiction to the definition of bridge. Hence e does not lie on a cycle.

Conversely, suppose e lies on no cycle in G.
Remove e from G, and we suppose there is a path between end vertices of e after removing e, as e is not a bridge. Now, add e again. As there is a path between two end vertices of e which is not through e, there is a cycle in G which is passed through e. Which is a contradiction. Therefore, e is a bridge of G.

Result: A connected graph T is a tree, iff every edge of T is a bridge.

Let T be a tree. Then T does not contain a cycle, and hence every edge is a bridge.

Conversely, let T is connected and every edge is a bridge. Then T is acyclic and connected$\Rightarrow$ T is a tree.

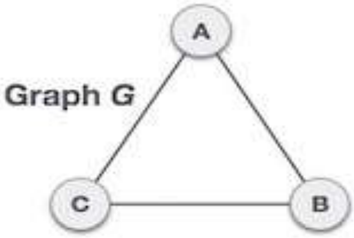
Spanning Tree:

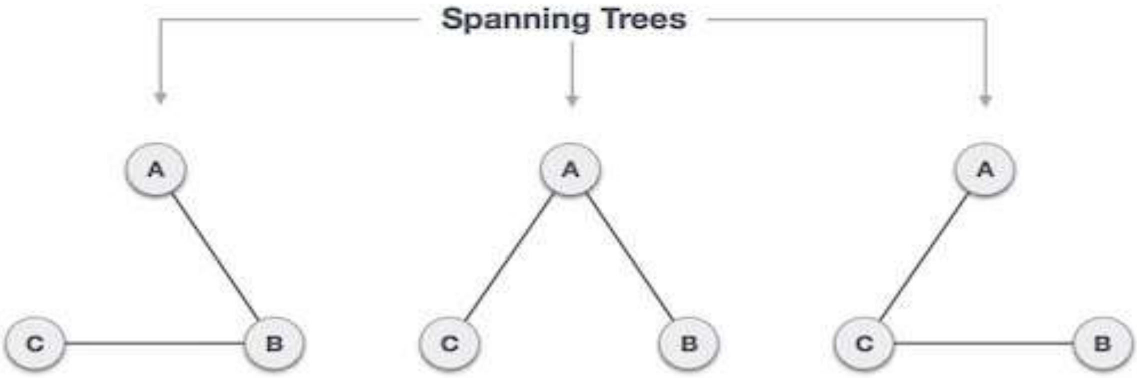
A tree T is said to be a ***spanning tree*** of a connected graph G, if T is a spanning subgraph of G.

Note: A connected graph may have several spanning trees.

Note: Each edge of a spanning tree is known as a ***branch***. Also an edge e in a connected graph G is a ***chord*** of the spanning tree T, if it is not an edge of T.
Clearly, in a graph G of order n and size m, there are $n-1$ branches and $m-n+1$ chords.

Example:





Note: for a complete graph of order n there are n^{n-2} number of spanning trees. Here $n=3$; hence number of spanning trees $=3^{3-2}=3$. This is known as the ***Cayley's formula***.

Result: Every connected graph G has at least one spanning tree.

Case1: If G has no circuit, it is itself a spanning tree.

Case2: If G has a circuit, we delete an edge from the circuit. This will still leave the graph connected. If there are more circuits, we continue this operation, till a an edge from the last circuit is deleted – leaving a connected, circuit-free graph that contains all the vertices of G. This is our required spanning tree.

Degree matrix:

This is defined for a graph G of order n as, $D(G) = (d_{ij})_{n\times n}$, where $d_{ij} = \begin{cases} d(v_i), if\ i = j \\ 0\ ,\ \ otherwise \end{cases}$

Laplacian matrix:

This is defined for a graph G of order n as, $L(G) = D(G) - A(G)$.

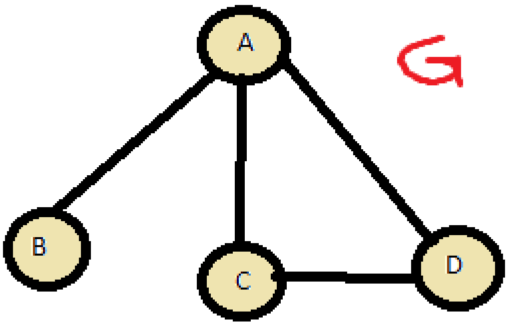
Kirchhoff's algorithm for finding number of spanning trees:

Step 1: Delete any loops from the given graph. Let the new graph be G.

Step 2: Create adjacency matrix $A(G)$ and degree matrix $D(G)$ and Laplacian matrix $L(G)$.

Step 3: Calculate co-factor for any element of $L(G)$. Its value gives the number of spanning trees of G.

Example:



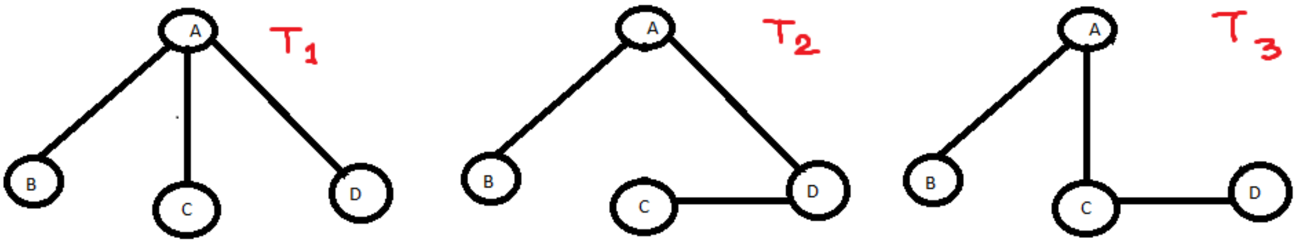
Let us try to find the number of spanning trees for the incomplete graph G:

Note: This algorithm can find the number of spanning trees for both complete as well as incomplete graphs, but for complete graph Cayley's formula gives result easily and quickly.

Here, $D(G) = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$; $A(G) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \Rightarrow L(G) = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{pmatrix}$

Cofactor of $L(G) = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 2 & -1 \\ 0 & -1 & 2 \end{vmatrix} = 3$

Hence, there are 3 possible spanning trees.



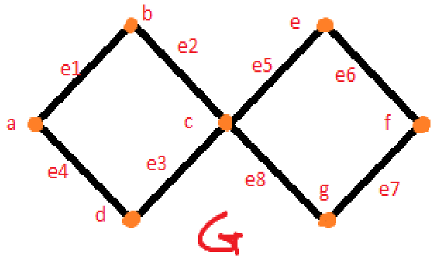
Note: If we add an edge between any two vertices of a tree a circuit is created. This is because there already exists a path between any two vertices of a tree; adding an edge between them therefore creates an additional path, and hence a circuit.

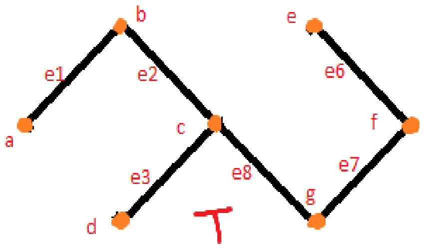
Thus, we can say, that ***a connected graph G is a tree iff adding an edge between any two vertices in G creates exactly one circuit***.

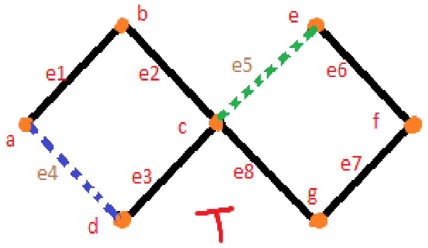
Fundamental Circuit:

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

Example:







Here G is a connected graph and T is a spanning tree.

With respect to T, there are two chords, e_4 and e_5 and hence there are two fundamental circuits $T \cup \{e_4\}$ and $T \cup \{e_5\}$

Note: Clearly the number of fundamental circuits would be equal to the number of chords, and hence for a graph with n vertices and m edges, the number of fundamental circuits is $= m - (n - 1) = m - n + 1$

In the above graph, $m = 8, n = 7$; hence number of fundamental circuit $= 8 - 7 + 1 = 2$.

Note: A circuit is a fundamental circuit only with respect to a given spanning tree. A given circuit may be fundamental with respect to one spanning tree, but not with respect to a different spanning tree of the same graph.

Rank & Nullity:

Let, G be a graph of order n and size m with k components.

Then $n \geq k$ and $m \geq n - k$ [$\because$ G is connected]

Thus, $n - k \geq 0$ & $m - n + k \geq 0$

The ***rank of a graph G***, denoted by r, is defined as $r = n - k \geq 0$.

The ***nullity of a graph G***, denoted by μ, is defined as $\mu = m - n + k \geq 0$

Note: If G is connected then obviously $k = 1 \Rightarrow r = n - 1$ & $\mu = m - n + 1$;

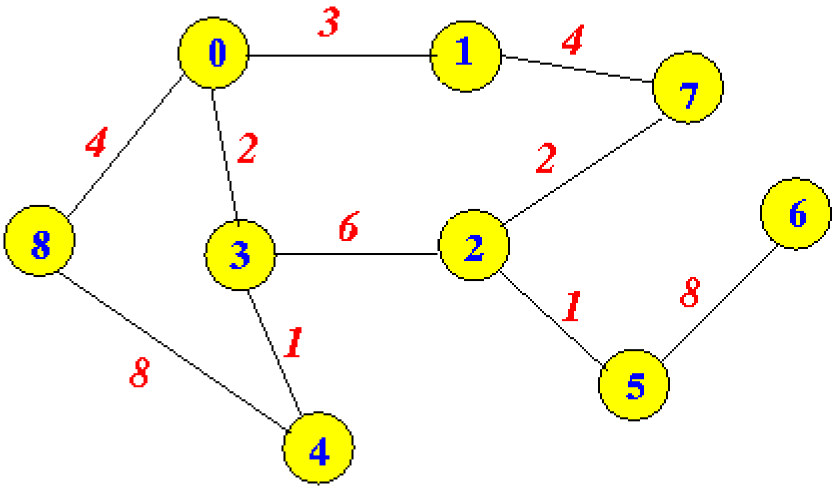
Hence, in case of a connected graph the rank r and the nullity μ denotes the number of branches and the number of chords respectively with respect to a given spanning tree.

Weighted Graph:

A graph G is said to be a weighted graph, if every edge of G is assigned a number that is known as the edge-weight.

Edge-weights are usually taken to be real numbers.

Example:



In the graph above weights of the sides {0,1}, {7,2}, {5,6} etc. are 3,2,8 etc.

The edge-weights usually represent:

a. ***Cost or Distance***: the **amount of effort** needed to **travel** from one place to another;
b. ***Capacity***: the **maximum amount of flow** that can be **transported** from one place to another.

Weight of a spanning tree:

Let T be a spanning tree of a connected edge-weighted graph G. Then the sum of all weights of the branches of T is known as the weight of the spanning tree T.

Minimal Spanning Tree (MST):

A spanning tree of G is said to be a minimal spanning tree of G if it has the smallest weight.

Note: There may be several minimal spanning trees having same total weight. But if the weights of edges are all different then there exists a unique minimal spanning tree. Again, if e be the only edge with minimum weight in G, it will be included in every minimal spanning tree of G.

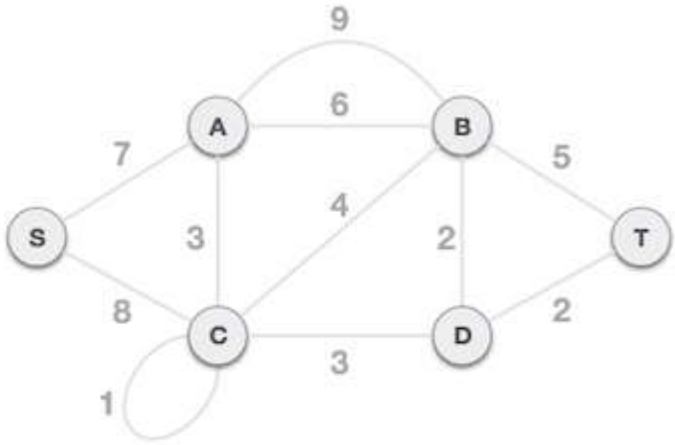
Minimal Spanning Forest:

If a weighted graph G is a disconnected with components $G_i, i = 1, . . , k$, each G_i has a minimal spanning tree, say T_i, such that $F = \cup_{i=1}^{k} T_i$, gives the minimal spanning forest of G.

Kruskal’s Algorithm:

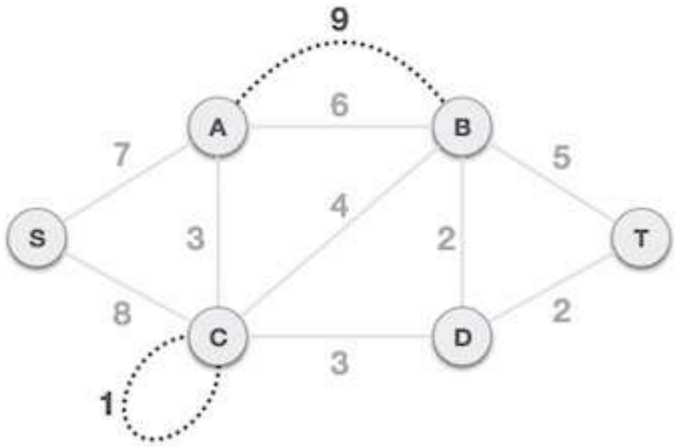
Kruskal's algorithm treats the graph as a forest and every node it has as an individual tree. A tree connects to another only and only if, it has the least cost among all available options and does not violate Minimum Spanning Tree properties.

To understand Kruskal's algorithm let us consider the following example:

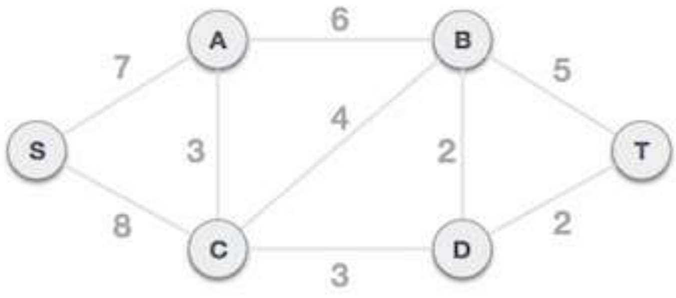


Step 1: Remove all loops and Parallel Edges

Removing all loops and parallel edges from the given graph we get,



In case of parallel edges, we keep the one which has the least cost associated and remove all others.



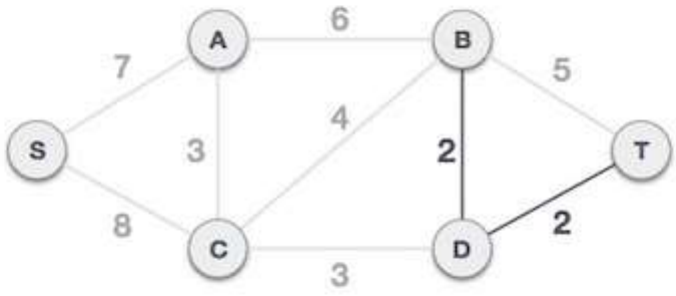
Step 2: Arrange all edges in their increasing order of weight

Arranging all the edges in an ascending order of weightage (cost).

B, D	D, T	A, C	C, D	C, B	B, T	A, B	S, A	S, C
2	2	3	3	4	5	6	7	8

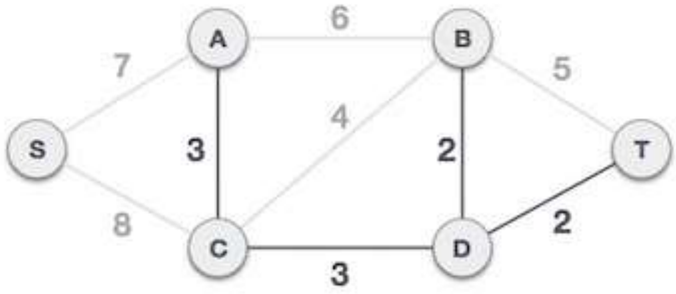
Step 3: Add the edge which has the least weightage

Now we start adding edges to the graph beginning from the one which has the least weight. Throughout, we shall keep checking that the spanning properties remain intact. In case, by adding one edge, the spanning tree property does not hold then we shall not include the edge in the MST.

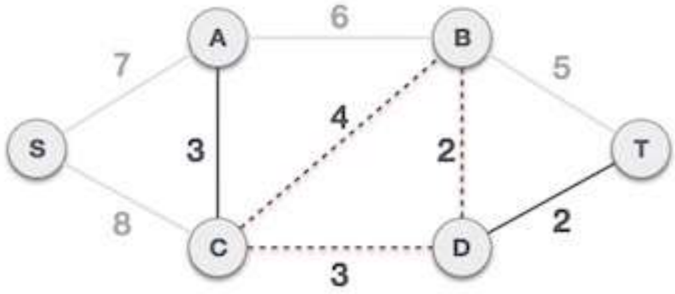


The least cost is 2 and edges involved are B,D and D,T. We add them. Adding them does not violate spanning tree properties, so we continue to our next edge selection.

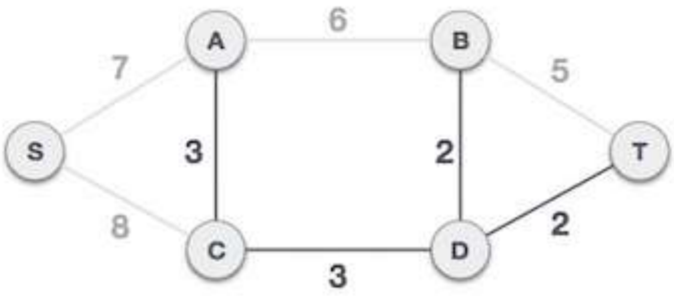
Next cost is 3, and associated edges are A,C and C,D. We add them again −



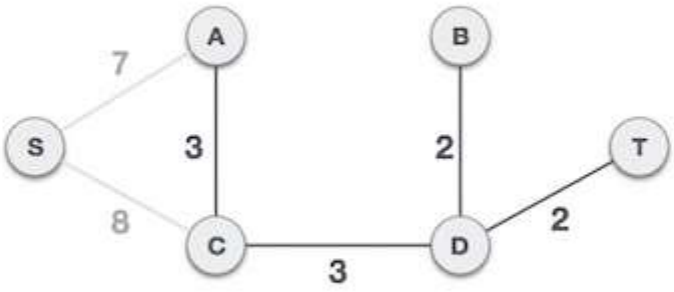
Next cost in the table is 4, and we observe that adding it will create a circuit in the graph. −



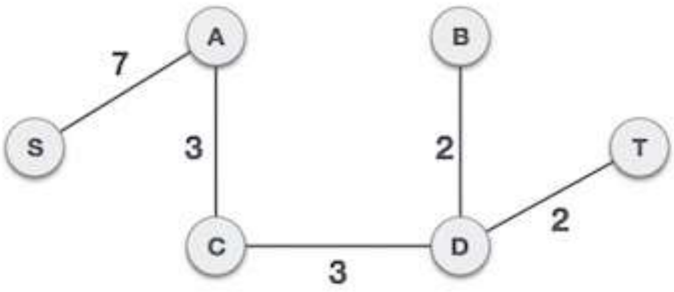
We ignore it. In the process we shall ignore/avoid all edges that create a circuit.



We observe that edges with cost 5 and 6 also create circuits. We ignore them and move on.



Now we are left with only one node to be added. Between the two least cost edges available 7 and 8, we shall add the edge with cost 7.

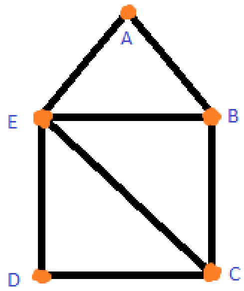


By adding edge S,A we have included all the nodes of the graph and we now have minimum cost spanning tree as given above with the minimum cost: $7 + 3 + 3 + 2 + 2 = 17$

Problems on Chapter 5:

1. Prove that a weighted connected graph with distinct weights has exactly one minimal spanning tree.

2. Find the number of spanning trees for the graph G below:



3. Let, G be a connected graph with order 6 and size 7. Show that there does not exist a spanning tree with 3 chords.

4. Show that a connected graph G of order 9, with 3 vertices of degree 4, 4 vertices of degree 3 and the remaining vertices of degree 1, cannot be a tree. Find rank and nullity of G.

5. Examine if a graph G with degree sequence $(1,1,2,8)$ is a tree. Does it have a spanning tree?

6. Let T be a tree with odd order. Show that T must have a vertex with even degree.

7. Show that a connected graph G of order $n(\geq 2)$ is a tree, iff $\sum_{v\in V} d(v) = 2n - 2$.

8. Examine whether a graph having no vertex of odd degree is a tree or not.

9. If T is a tree of order $n(\geq 2)$ and n_1 be the number of vertices with degree 1 and p is the number of vertices of degree greater or equal to three, show that $n_1 \geq p + 2$.

10. Five villages A, B, C, D, E, are to be connected by pipelines. Cost of connection in units of 10,000 ₹ is given in the following table. Find the total minimum cost.

	A	B	C	D	E
A	----	2	4	3	5
B	2	----	7	4	6
C	4	7	----	10	8
D	3	4	10	----	9
E	5	6	8	9	----

Hints & Answers:

1. ***Assumption:***
   In a graph with distinct weights, there are two MSTs. S and T.
   Proof by contradiction:
   Let us assume an edge with lowest weight e=PQ which is present in S but not in T.
   If we add e to T, we will have a cycle. If all other edges in T were in S, S would definitely have a cycle, which it cannot.
   So, the cycle in T must contain an edge f which is not in S.
   So, by definition of e and the fact that all weights are different gives W(e)<W(f).
   So, if we replace f by e we get a MST with smaller cost which is a contradiction.
   Hence, S and T must be the same.

2. The graph G is not complete. Here we have,
$$D(G)=\begin{pmatrix}2&0&0&0&0\\0&3&0&0&0\\0&0&3&0&0\\0&0&0&2&0\\0&0&0&0&4\end{pmatrix};A(G)=\begin{pmatrix}0&1&0&0&1\\1&0&1&0&1\\0&1&0&1&1\\0&0&1&0&1\\1&1&1&1&0\end{pmatrix}\Rightarrow L(G)=\begin{pmatrix}2&-1&0&0&-1\\-1&3&-1&0&-1\\0&-1&3&-1&-1\\0&0&-1&2&-1\\-1&-1&-1&-1&4\end{pmatrix}$$
$$\text{Cofactor of } L(G)=\begin{vmatrix}3&-1&0&-1\\-1&3&-1&-1\\0&-1&2&-1\\-1&-1&-1&4\end{vmatrix}=21;$$
   hence, by Kirchhoff's formula there are a total of 21 spanning trees of G.

3. For a connected graph G we must have $m-n+1$ chords, where m =number of edges; n =number of vertex;
   Here, $n=6\,;m=7\Rightarrow m-n+1=7-6+1=2$
   Thus, there does not exist any spanning tree with 3 chords.

4. Here $n=9$; 3 vertices of degree 4, 4 vertices of degree 3 and $(9-3-4)=2$ vertices of degree 1
   Total degree of all vertices $=3\times4+4\times3+2\times1=26$;
   Hence total edges $=\frac{26}{2}=13$
   But as $n=9$, any tree will have $9-1=8$ edges, and hence G is not a tree.
   Rank of $G=n-1=9-1=8$; Nullity of $G=m-n+1=13-9+1=5$.

5. Total degree $=1+1+2+8=12\Rightarrow m=\frac{12}{2}=6$
   but as $n=4$, any tree must have $4-1=3$ edges and hence G is not a tree.
   Again any graph G which is connected must have a spanning tree.
   Also, a graph G of order n is connected if $m\geq n-1$. Here $m=6>n-1=3$
   Hence, G is connected and therefore must have a spanning tree.

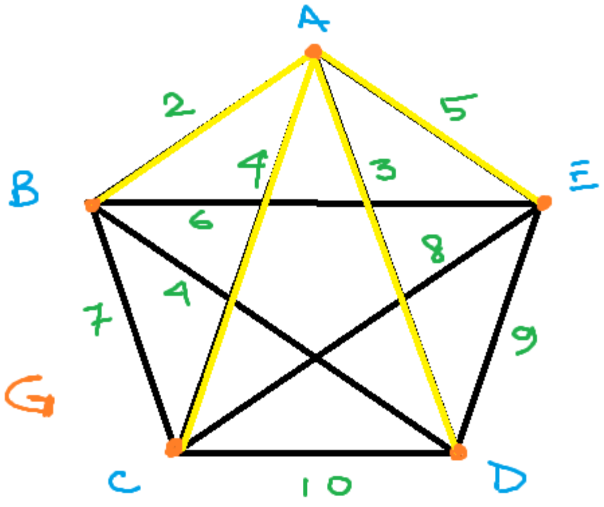
6. ***Proof by contradiction:***
   Let all the vertices be of odd degree.
   Since T is a tree with odd order⇒ n is odd⇒total degree of all vertices is also odd!
   clearly a contradiction, as sum of all degrees of vertices is always even.
   Hence the result.

7. Here, $\sum_{v \in V} d(v) = 2n - 2 = 2m \Rightarrow m = n - 1$
   And we know that a connected graph of degree n is a tree iff the number of edges $m = n - 1$.
   hence the result.

8. Since in a tree there are at least two pendant vertices (i.e. vertex with degree 1)
   hence a graph having no vertex of odd degree is definitely not a tree.

9. Let n_i be the number of vertices with degree i in T.
   Then, order of T is, $n = \sum n_i$ and size of T is, $m = \frac{1}{2}\sum i. n_i$
   Since T is a tree, we must have, $m = n - 1 \Rightarrow \frac{1}{2}\sum i. n_i = \sum n_i - 1 \Rightarrow 2\sum n_i = \sum i. n_i + 2$
   Again, $\sum i. n_i \geq n_1 + 2. n_2 + 3(n_3 + n_4 + \cdots) \Rightarrow 2\sum n_i \geq n_1 + 2. n_2 + 3(n_3 + n_4 + \cdots) + 2$
   Hence, $2. n_1 + 2. n_2 + 2(n_3 + n_4 + \cdots) \geq n_1 + 2. n_2 + 3(n_3 + n_4 + \cdots) + 2$
   i.e. $n_1 \geq (n_3 + n_4 + \cdots) + 2$; or, $n_1 \geq p + 2$

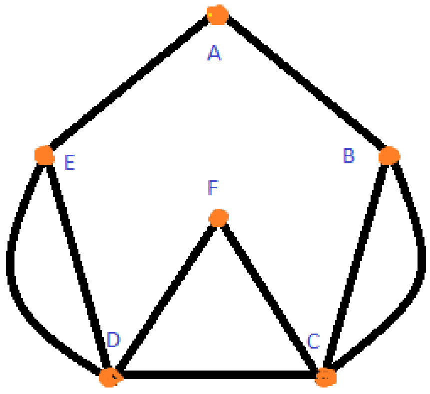
10. This problem of finding minimum cost is the same as to find a MST of G with G given below:



The non-decreasing sequence of weights of all the edges is {2,3,4,4,5,6,7,8,9,10}
The smallest element is 2 for AB, and so we add AB.
The next smallest is 3 for AD, and so we also add AD.
The next smallest is 4, for AC and BD. We cannot join BD because it creates a circuit with AB and AD. So we connect AC.
Next element is 5 for AE and we add AE.
Since the number of vertices is 5 hence any spanning tree will have exactly 5-1=4 edges.
So we have obtained our minimal spanning tree with edges AB, AC, AD and AE
The total minimum cost therefore is (2+4+3+5) × 10,000 ₹ = 14 × 10,000 ₹ =1,40,000 ₹

Assignments on Chapter 5:

1. Find two spanning trees of the graph G, given below. Also find the branches and chords of the spanning trees. Again, find all fundamental circuits for any one of the spanning trees.



2. Show that for a complete graph $K_n, r = \frac{2\mu}{n-2}$, where r, μ denote the rank and nullity of K_n.

3. Find rank and nullity of the complete bipartite graph $K_{2,3}$. Also find the number of spanning trees.

4. A forest F has 21 vertices and 3 components. Find the number of edges of F.

5. Show that a Hamiltonian path is a spanning tree.

6. Show that the complete bipartite graph $K_{m,n}$ cannot be a tree for $m, n > 1$.

7. Show that a tree of order $n \geq 2$ is a bipartite graph.

8. Let A, B, C, D, E, F be six cities, that are to be connected by a network of roads. The table below gives the distances in unit of 20 km. between these cities. Find the minimum length of the road network that connects all of them (Using Kruskal's algorithm).

	A	B	C	D	E	F
A	-	6	5	4	10	4
B	6	-	2	3	5	7
C	5	2	-	4	3	6
D	4	3	4	-	2	5
E	10	5	3	2	-	3
F	4	7	6	5	3	-