MATHEMATICAL FOUNDATION FOR COMPUTER SCIENCE

Prepared by: Er. Ankit Kharel

Nepal college of information technology

GRAPH THEORY

TREES:

- Tree is a connected undirected graph with no simple circuits
- Because a tree cannot have a simple circuit, a tree cannot contain multiple edges or loops. Therefore any tree must be a simple graph.

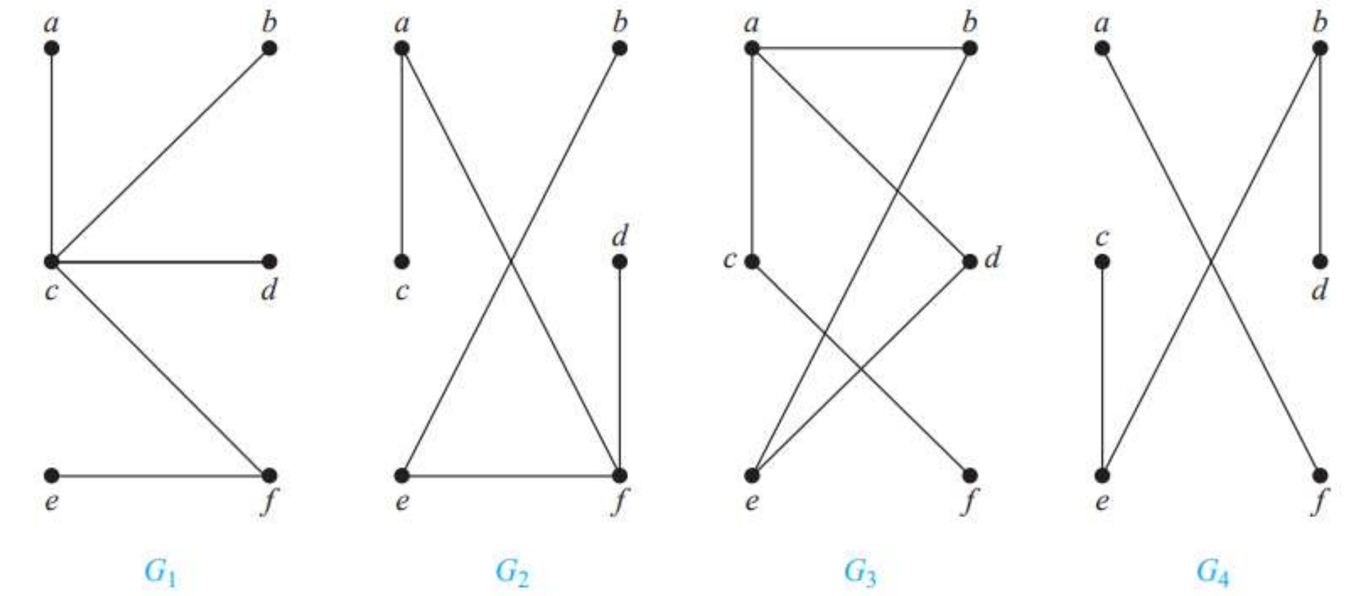


FIGURE 2 **Examples of Trees and Graphs That Are Not Trees.**

- G1 and G2 are trees
- G3 is not a tree because e, b, a, d, e is a simple circuit in this graph. Finally, G4 is not a tree because it is not connected.

ROOTED TREES:

- A rooted tree is a tree in which one vertex has been designated as the root and every edge is directed away from the root.

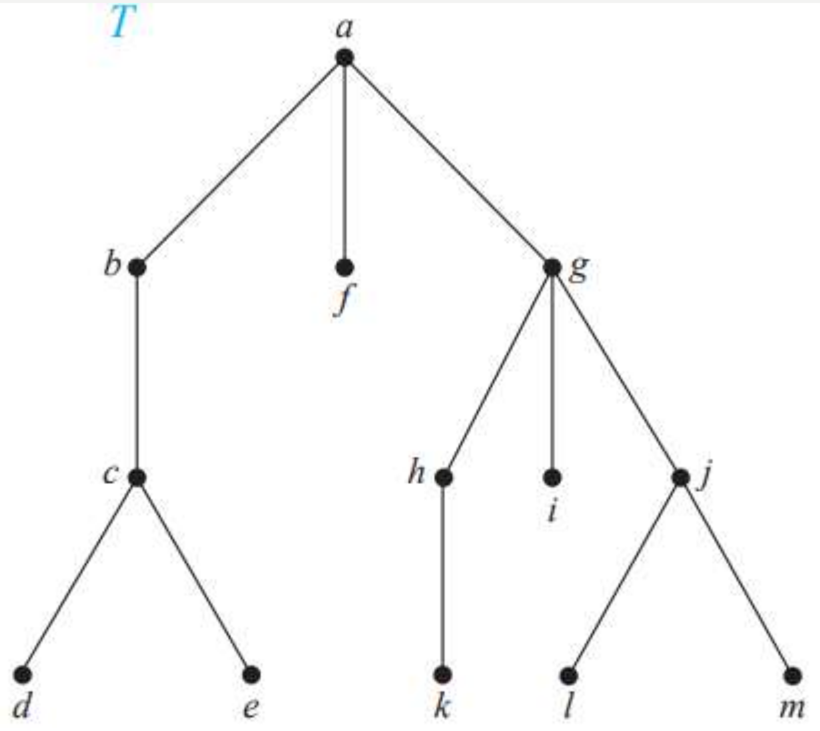
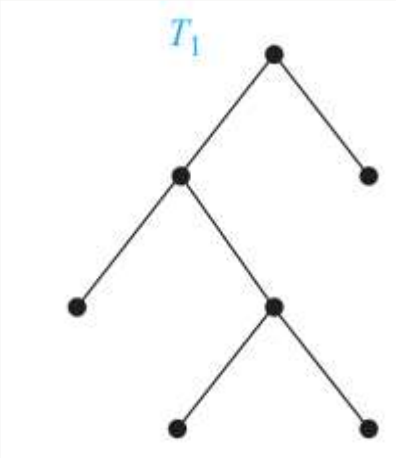


FIGURE 5 A Rooted Tree T.

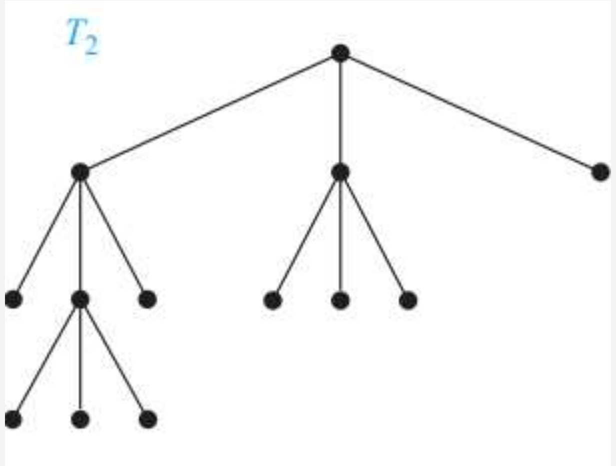
- A vertex of a rooted tree is called a **leaf** if it has no children.
- Vertices that have children are called **internal vertices**.
- Vertices with the same parent are called **siblings**
- The **ancestors** of a vertex other than the root are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root.
- The **descendants** of a vertex v are those vertices that have v as an ancestor

ROOTED TREES:

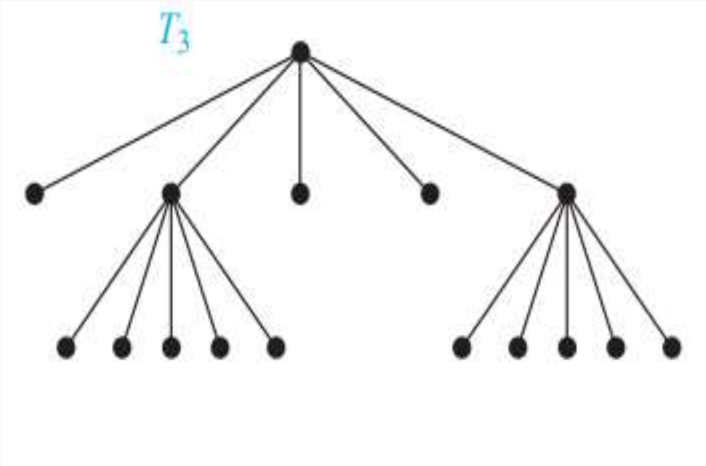
- A rooted tree is called an **m-ary** tree if every internal vertex has no more than m children. The tree is called a **full m-ary** tree if every internal vertex has exactly m children. An m-ary tree with m = 2 is called a binary tree.



T1 is a full binary tree because each of its internal vertices has two children



T2 is a full 3-ary tree because each of its internal vertices has three children

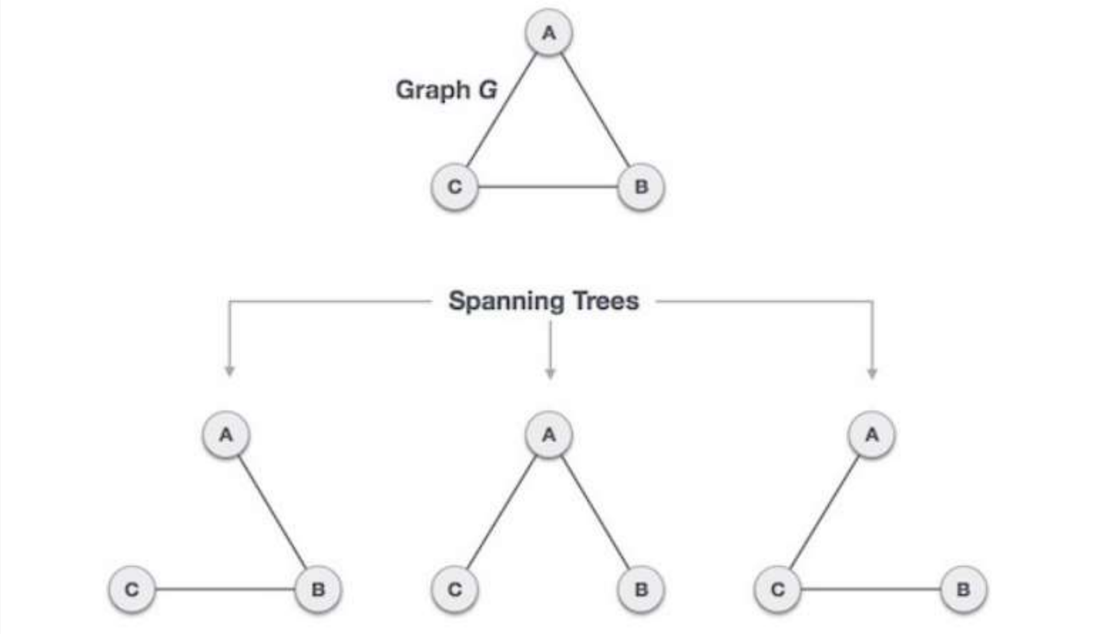


In T3 each internal vertex has five children, so T3 is a full 5-ary tree

SPANNING TREE:

- A **spanning tree** is a subset of Graph G, which has all the vertices covered with minimum possible number of edges. Formally, Let G be a simple graph. A spanning tree of G is a subgraph of G that is a tree containing every vertex of G.

- Every connected and undirected Graph G has at least one spanning tree. A disconnected graph does not have any spanning tree, as it cannot be spanned to all its vertices.



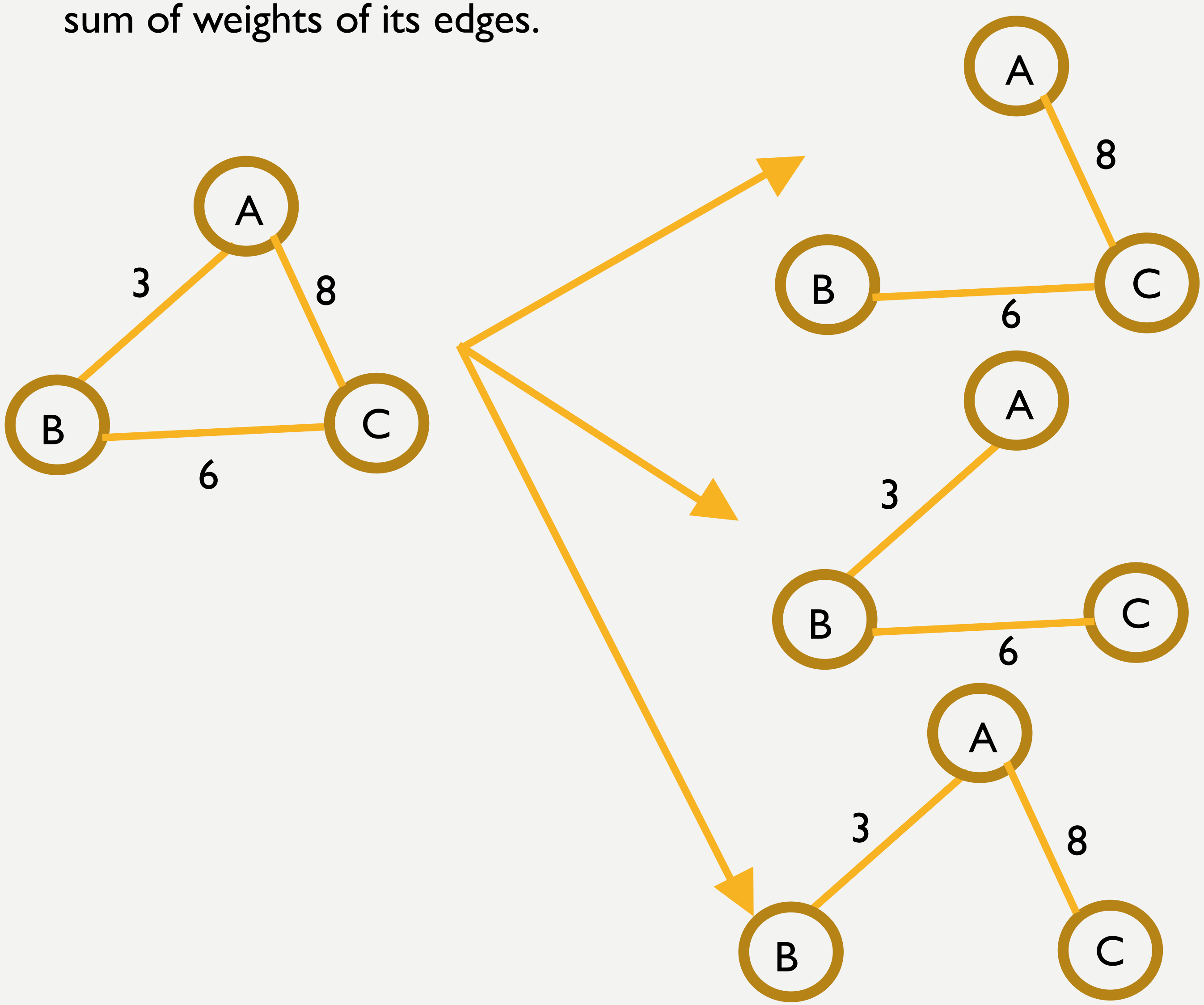
A complete undirected graph can have maximum $\mathbf{n^{n-2}}$ number of spanning trees, where **n** is the number of nodes. In the above addressed example, **n is 3,** hence $\mathbf{3^{3-2} = 3}$ spanning trees are possible.

SPANNING TREE:

- **General Properties of Spanning Tree**
  - ✓ A connected graph G can have more than one spanning tree.
  - ✓ All possible spanning trees of graph G, have the same number of edges and vertices.
  - ✓ The spanning tree does not have any cycle (loops).
  - ✓ Removing one edge from the spanning tree will make the graph disconnected, i.e. the spanning tree is **minimally connected**.
  - ✓ Adding one edge to the spanning tree will create a circuit or loop, i.e. the spanning tree is **maximally acyclic**.
  - ✓ Spanning tree has **n-1** edges, where **n** is the number of nodes (vertices).
  - ✓ A complete graph can have maximum n^{n-2} number of spanning trees.

MINIMUM SPANNING TREE(MST:

- A minimum spanning tree in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges.



Total cost = 14

Total cost = 9
Minimum Spanning Tree

Total cost = 11

MINIMUM SPANNING TREE(MST:

- There are two algorithms for finding Minimum Spanning Tree:
  - (a) Kruskal's Algorithm
  - (b) Prim's Algorithm

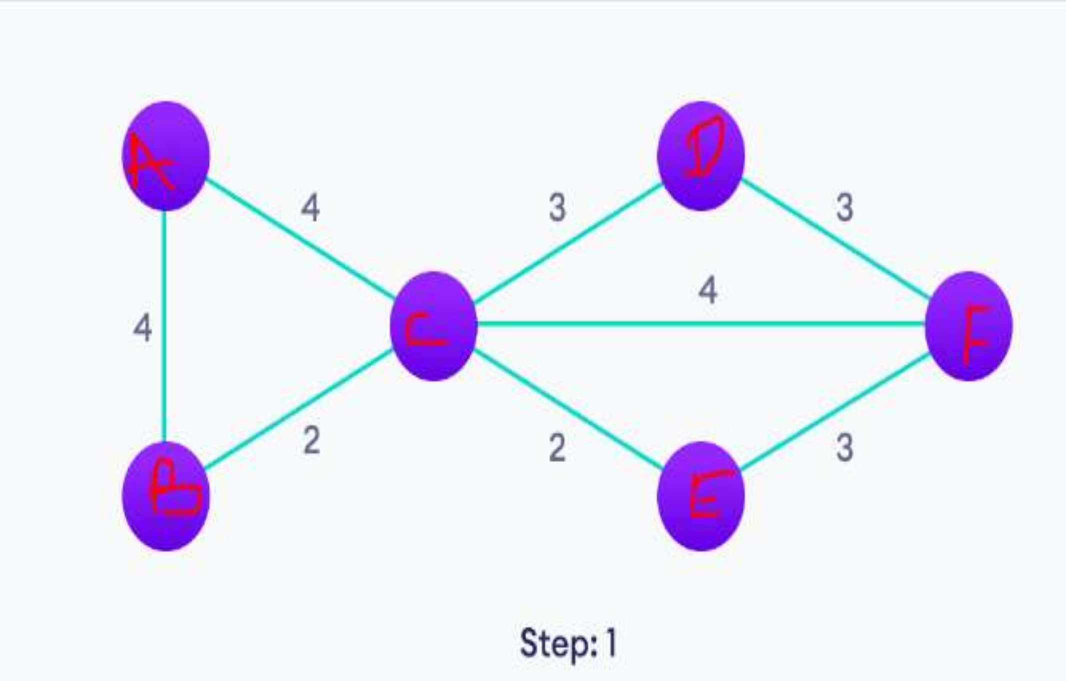
(a)KRUSKAL's ALGORITHM:

The steps for implementing Kruskal's algorithm are as follows:

- Remove all the loops and parallel edges if present. In case of parallel edges, keep the one which has the least cost associated and remove all others.
- Sort all the edges from low weight to high
- Take the edge with the lowest weight and add it to the spanning tree. If adding the edge created a cycle, then reject this edge.
- Keep adding edges until we reach all vertices.

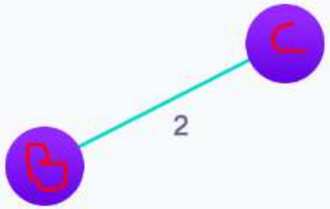
MINIMUM SPANNING TREE(MST:

Find The minimum spanning tree using Kruskal's Algorithm.



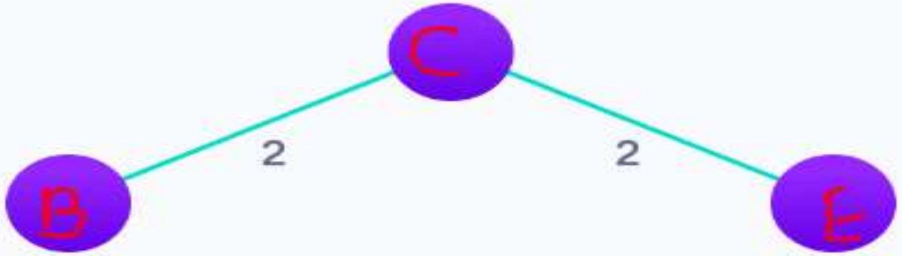
STEP: I Sort all the edges in ascending order

{B ,C}	{C ,E}	{D ,C}	{E ,F}	{D ,F}	{B ,A}	{A ,C}	{F ,C}
2	2	3	3	3	4	4	4



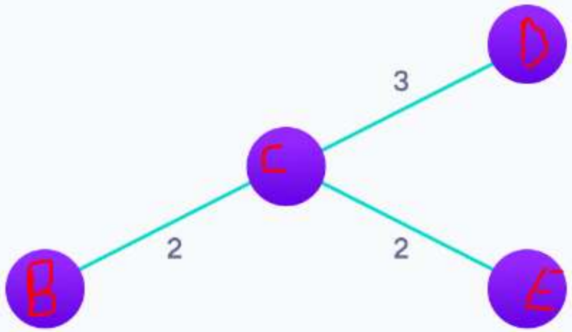
Step: 2

Choose the edge with the least weight, if there are more than 1, choose anyone



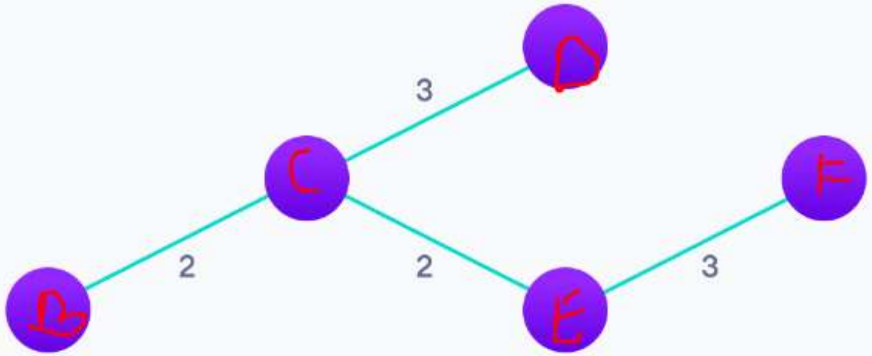
Step: 3

Choose the next shortest edge and add it



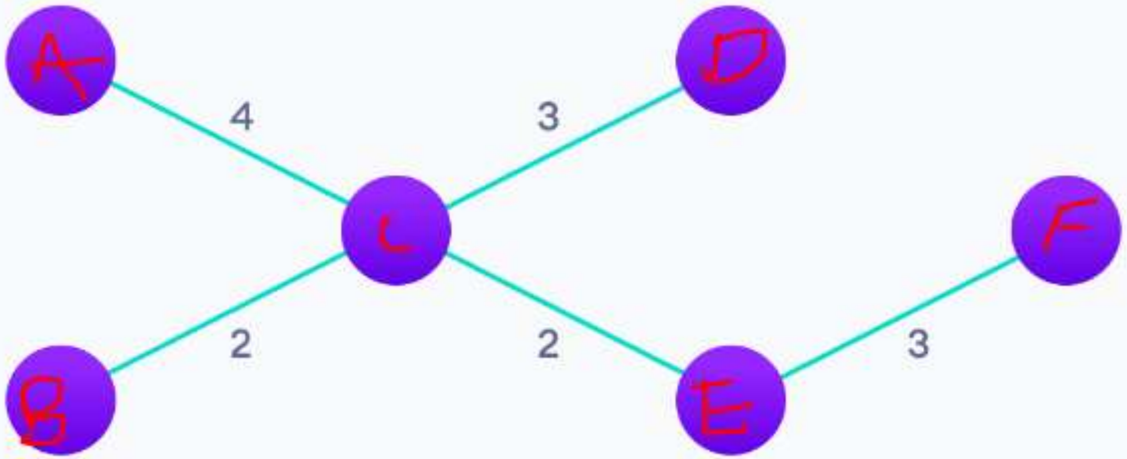
Step: 4

Choose the next shortest edge that doesn't create a cycle and add it



Step: 5

Choose the next shortest edge that doesn't create a cycle and add it

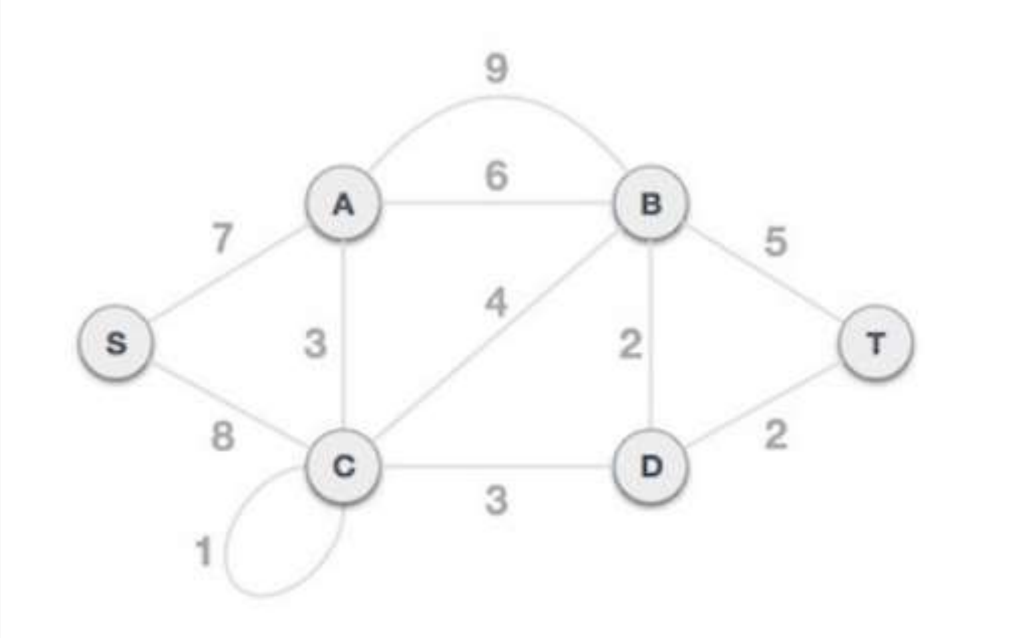


Step: 6

Repeat until you have a spanning tree

MINIMUM SPANNING TREE(MST:

Find The minimum spanning tree using Kruskal's Algorithm.

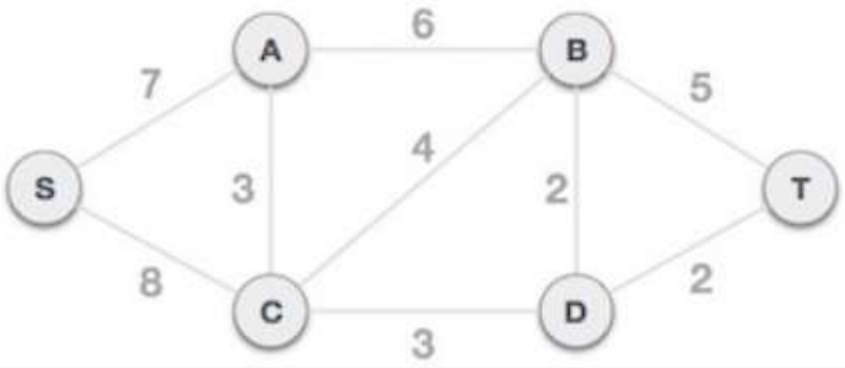


STEP: I Remove loops and parallel edges:

{B ,C}	{C ,E}	{D ,C}	{E ,F}	{D ,F}	{B ,A}	{A ,C}	{F ,C}
2	2	3	3	3	4	4	4

MINIMUM SPANNING TREE(MST:

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



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

The next step is to create a set of edges and weight, and arrange them 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

MINIMUM SPANNING TREE(MST:

- There are two algorithms for finding Minimum Spanning Tree:
  - (a) Kruskal's Algorithm
  - (b) Prim's Algorithm

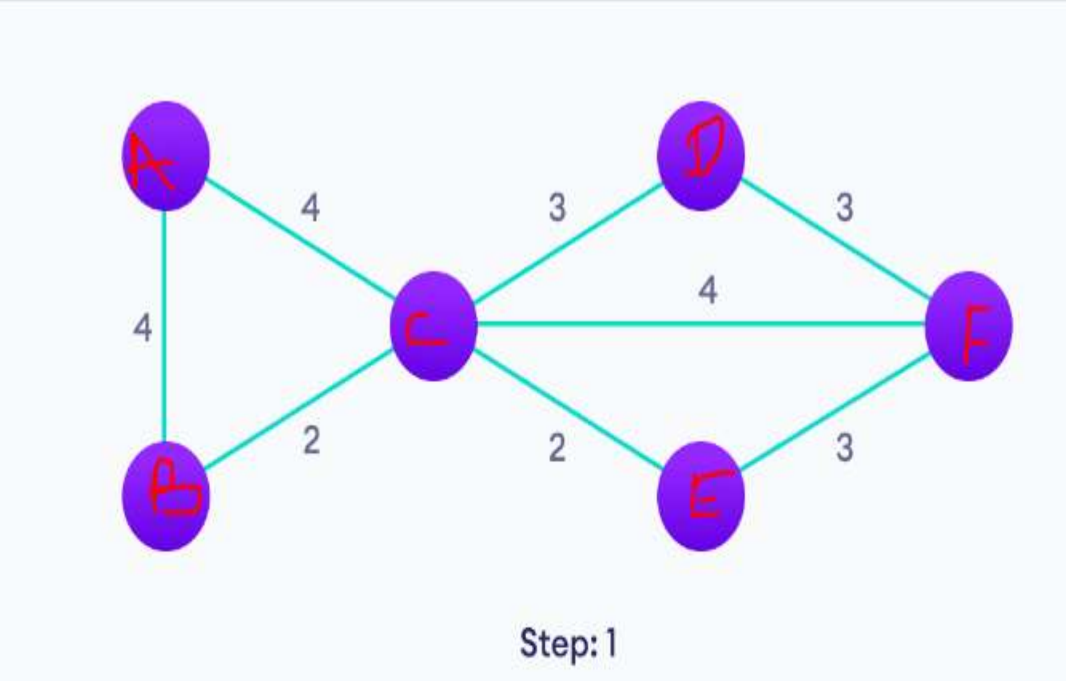
(a)PRIM's ALGORITHM:

The steps for implementing Prim's algorithm are as follows:

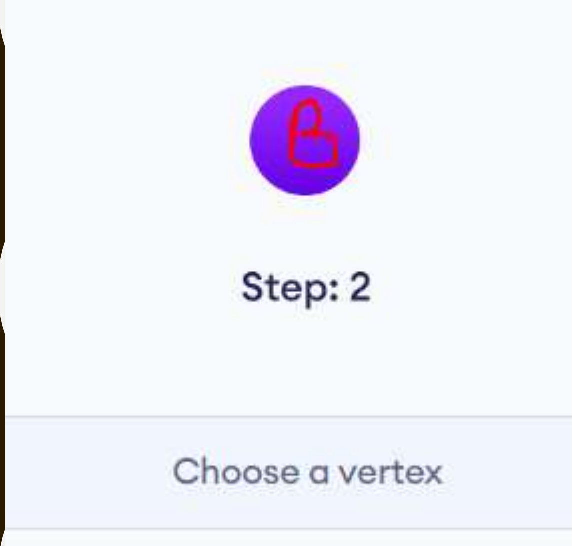
- Remove all the loops and parallel edges if present. In case of parallel edges, keep the one which has the least cost associated and remove all others.
- Choose any arbitrary vertex as a root
- Check outgoing edges and select the one with least cost and no cycle
- Repeat step(ii) until all vertices are covered.

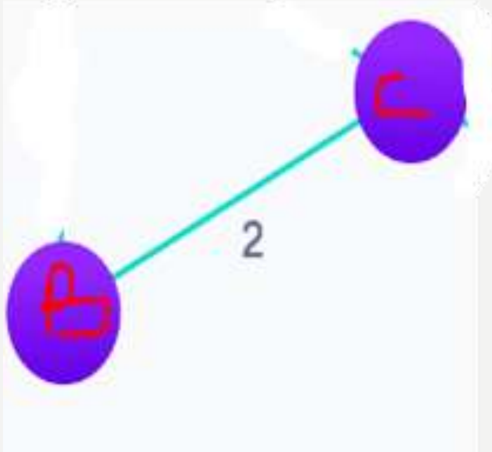
MINIMUM SPANNING TREE(MST:

Find The minimum spanning tree using Prim's Algorithm.

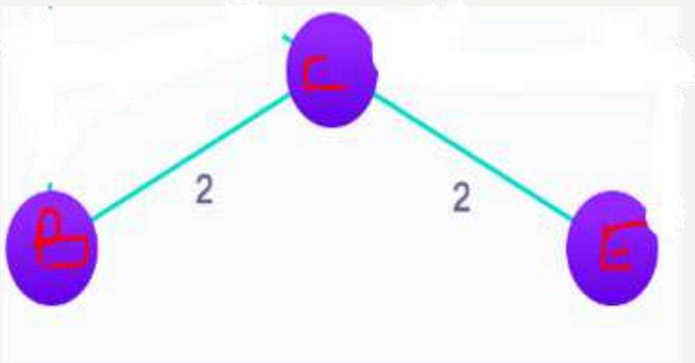


Step: 1 If loops and parallel edges are present remove it

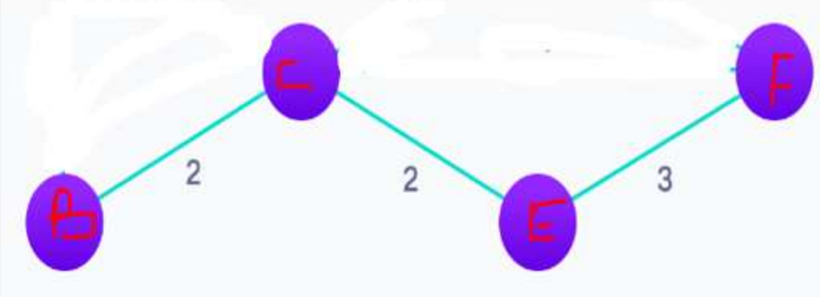




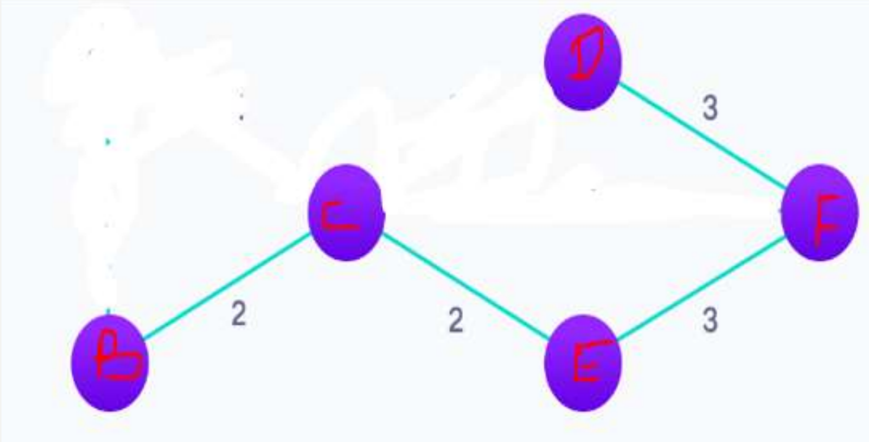
Step: 3 Choose the smallest weight connected edge



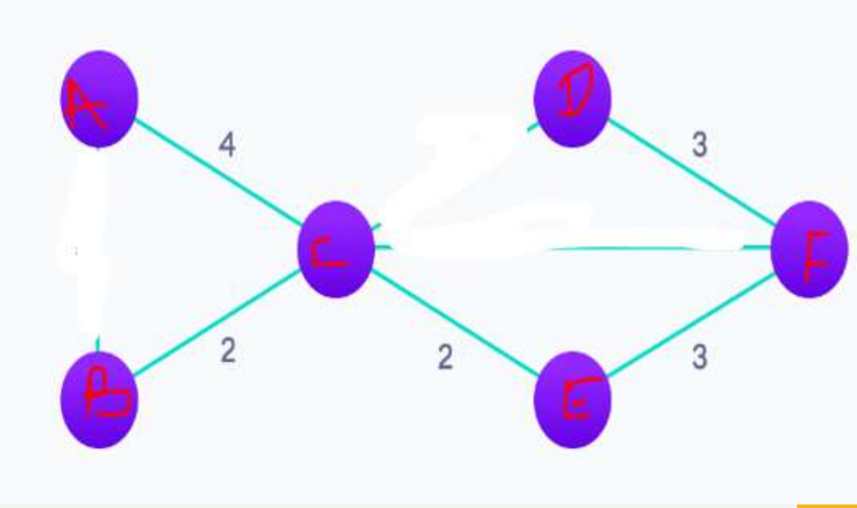
Step: 4



Step: 5



Step: 6



Step: 7
MST
Weight = 14

MINIMUM SPANNING TREE(MST:

Find The minimum spanning tree using Kruskal's Algorithm.