Assignment Unit-II

Q1. What is tree and Forest? Explain it's characteristic.

TREE

In computer science, a tree is a hierarchical data structure that consists of nodes connected by edges. A tree start with a root node and expands into child nodes, resembling a tree-like structure. It's widely used for representing hirarchical data, such as file systems, organizational structures, or decision process.

Characteristics of a Tree.

1. Root Node: The topmost node in the tree, Every tree has exactly one root node.
2. Nodes: Individual elements in the tree, which store data. They are connected by edges.
3. Parent - Child Relationship: Every node (except the root) has a parent, and nodes can have multiple children.
4. Leaf Nodes: Node that do not have any children.
5. Edges: The connections btw nodes.
6. Depth: The number of edge on the longest path from the root to a leaf.
7. Depth: The number of edge from the root to a particular node.
8. Subtree: A tree formed by a node and its descendants.

3. Binary Tree: A special kind of tree where each node has at most two children (left and right)

FOREST

A forest is a collection of disjoint tree. In simpler terms, a forest is just a set of multiple independent tree that do not share any connection. Each tree in a forest has its own root node and exists as a seperate entity.

Characteristics of a Forests

1. Multiple Root Nodes: Unlike a tree, a forest can have more than more than one root node, as it consists of multiple independent trees.
2. Disjoint tree: The tree in a forest are seperate from each other and do not shar any nodes.
3. Transformation to a Tree: A forests can be con verted into a tree by adding a new root node and connecting it to the root nodes of the trees in the forests
4. Used in Graph Theory: In graph theory, a forest is considered a special type of graph with no cycles.

Q2. What is spanning tree? Write down the method of finding the spanning tree with example.

Spanning Tree

A spanning tree is a subset of a graph that include all the vertices (nodes) of the original graph but only enough edges (connection) to form a tree. A spanning tree covers all vertices without forming any cycles, meaning there is exactly one path btw any two vertices. This concept is mostly used in the context of Graph theory and is widely applied in network of deg design, communication system, and circuit design.

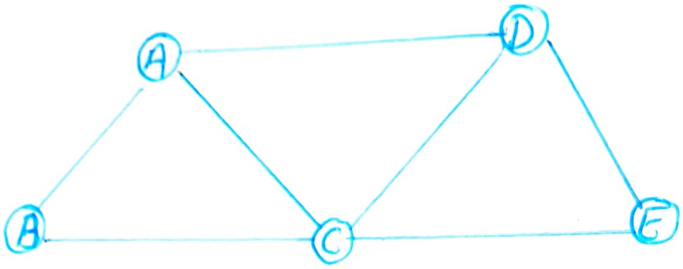
There are two methods of finding the spanning tree.

i) Cutting Down Method
ii) Building Up Method.

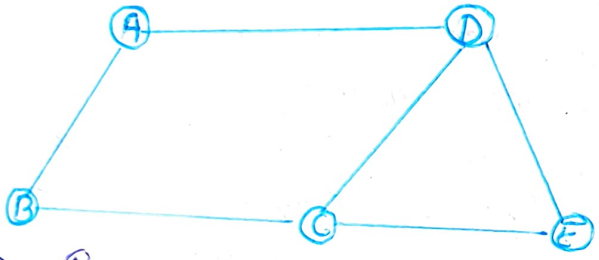
i) Cutting Down Method.

Step-1 -> Start choosing any cycle in Graph 'G'
Step-2 -> Removes one of cycle's edge.
Step-3 -> Repeat this process untill there are no cycle left.

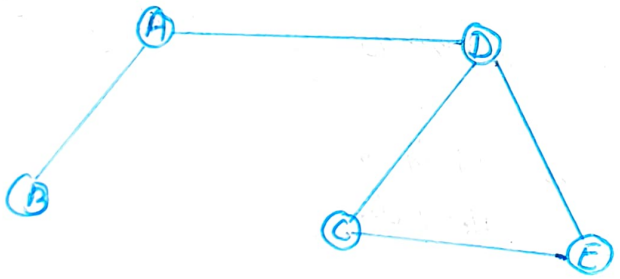
Ex- If there are Graph 'G'



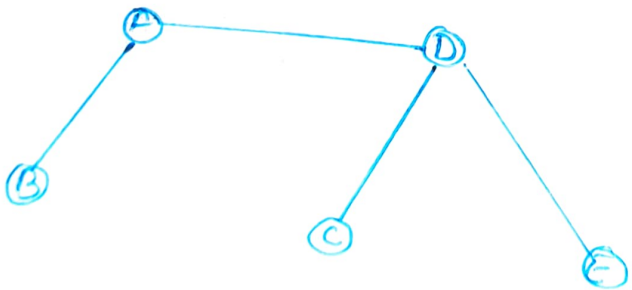
Step-4 → If we remove the edge a, c, which distroy the cycle a, d, c, a in the above graph then we get the following graph.



Step-5 → Remove the edge c, d which distroy the cycle A, B, C, A from the above graph.



Step-6 → Remove edge E, C from the cycle D, E, C, D from the above graph, then we get the following spanning tree.



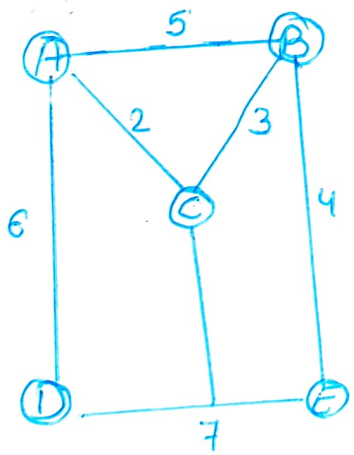
Que 3. Define maximum & minimum spanning tree with example.

Minimum Spanning Tree (MST)

A Minimum Spanning Tree (MST) of a weighted, connected graph is a subset of edge that connect all the vertices together without any cycle and with the minimum possible total edge weight.

Key Characteristics.

- Connected all vertices of the graph.
- No cycles.
- Minimizes the sum of weight of its edge.



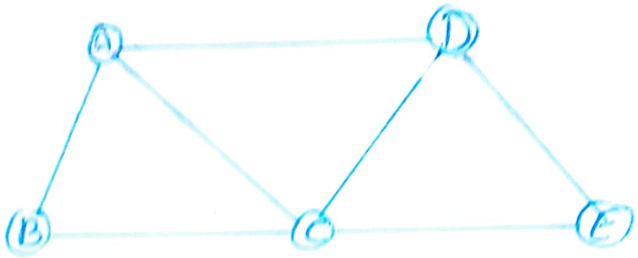
- Vertices : A, B, C, D, E
- Edges with weight AB = 5, AC = 2, BC = 3, BD = 6
  BE = 4, DE = 7

11) Building Up Method :-

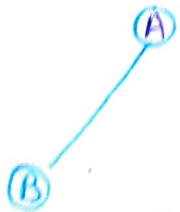
Step-1 :- Select edges of Graph 'G' one at a time in such a way that there are no cycles are created.

Step-2 :- Repeat this process untill all the vertices are included.

ex - Graph 'G'



- Choose the edge A, B



- Choose the edge D, E

7 - (5-1)
9 - (6-1)

- Choose the edge C, E

- Choose the edge C, B

- finally we got the spanning tree

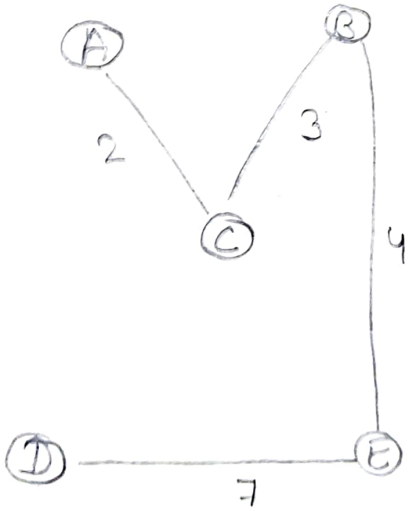
Now, we find the MST:

1. Start with the edge with the lowest weight AC (weight = 2).
2. Next, pick BC (weight = 3) because it connects C and B without forming a cycle.
3. Then, pick BE (weight = 4).
4. Finally, pick DE (weight = 7). The MST doesn't include AB or BD, as those would form cycles.

The MST is formed with edges:

AE, BC, BE, DE

The total weight is $2+3+4+7=16$

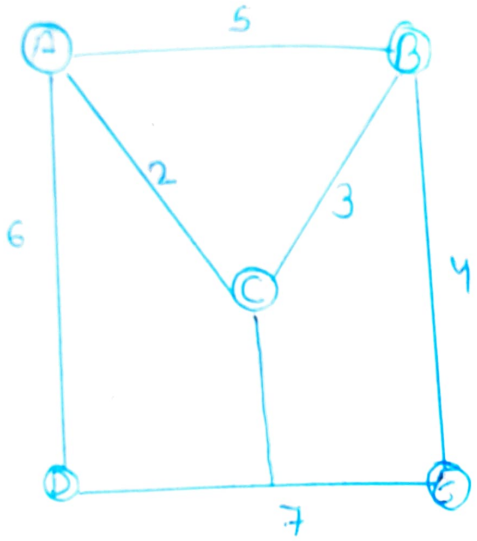


Final Spanning Tree

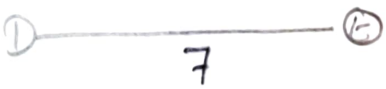
Maximum Spanning Tree

A maximum Spanning Tree is similar to a minimum spanning Tree but instead of minimizing it maximize the total edge weight

Example:



1. Start with the largest edge DE (weight = 7)



2. Then pick BD (weight = 6)

Q4. Define the shortest path explain Dijistra's Algorithm.

Shortest Path

Shortest path in a graph is the path blw two vertices has the minimum total weight (or distance of edges. It is especially useful in weighted graph where edge has a cost associated with it, such as road networks, computer networks, or transit system.

Dijistra's Algorithm

Dijistra's Algorithm is a popular algorithm for solving many single source shortest path problem, having non negative edge weight in the graph such that it is to find shortest distance between to vertices on a graph. It was conceived by dutch computer scientist "Edgser W Dijistra's in 1956.

The Algorithm maintains the set of visited vertices and a set of unvisited vertices it starts at the sources vertex and selects the unvisited vertex with the smallest tantative vertex from the sources. If then visit the neighbor of these vertex and updates this tantative distance if a shortest path is found. This process continue untill the destination vertex is reached

or all reachable vertices have been visited.

Steps for Dijkstra's Algorithm

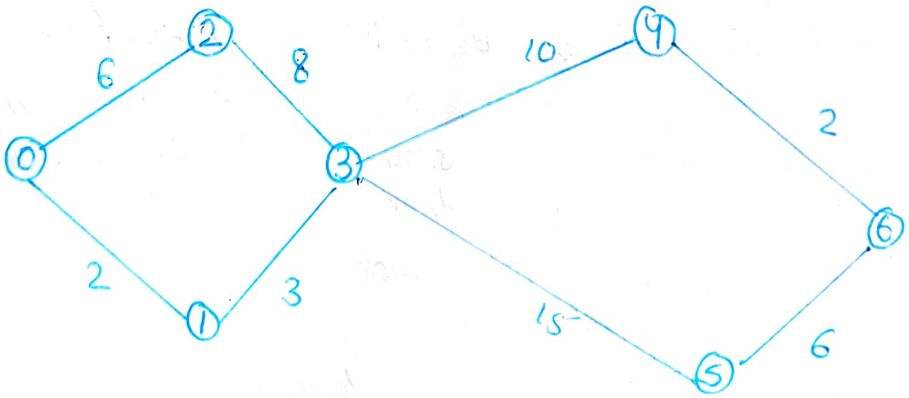
Step-1 → Mark the source node with a current distance of zero and the rest with infinite.

Step-2 → Set the non-visited node with the smallest current distance as the current node.

Step-3 → For each neighbor N of the current node adds the current distance of the adjacent node with the weight of the edge connecting 0 → 1. If it is smaller then the current distance of Node, set it as the new current distance of N.

Step-4 → Mark the current node 1 as visited. go to Step-2 if there are any nodes are unvisited.

Example :-



Dijistra's Algorithm will generate the shortest path node 0 → 1 to all other nods in the graph. As we can see we have the shortest path from

Node 0 to node 1, from
Node 0 to node 2, from
Node 0 to node 3, from
Node 0 to node 4, from
Node 0 to node 6, from.

Initially we have a set of resources given below.

i) The distance from the source node to it's self is 0.
In this ex. the we have source node is 0.

ii) This distance from the source node to all other node is unknown so we maker all of them as ∞.

0 → 0, 0 → 1, 0 → 2, 0 → ∞, 0 → ∞,
0 → ∞, 0 → ∞.

iii) Will also have an array of unvisited element that will keep track of unvisited or unmarked nodes.

iv) Algo. will computer when all the nodes marked as visited and the distance b/w them added to the path.
Initially unvisited node is 0, 1, 2, 3, 4, 5, 6

To visit all the nodes :-

Step-1. Start from 0 and mark node as visited.

Visited node = 0

Unvisited node = 1, 2, 3, 4, 5, 6

Distance from node 0

$0 \rightarrow 1 = 2$

$0 \rightarrow 2 = 6$

$0 \rightarrow 3 = \infty$

$0 \rightarrow 4 = \infty$

$0 \rightarrow 5 = \infty$

$0 \rightarrow 6 = \infty$