Trees...

Learn Vid Fun...

Definitions of Tree:

Tree is a non-linear data structure which organizes data in hierarchical structure and this is a recursive definition.

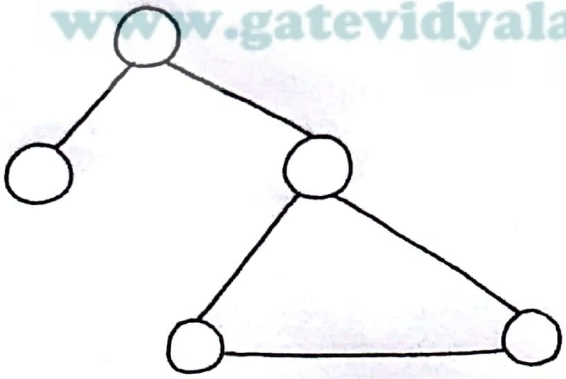
OR

A tree is a connected graph without any circuits.

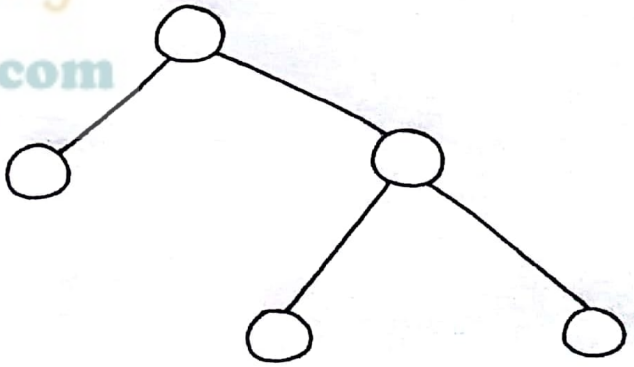
OR

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

Example:



Graph which is not a tree

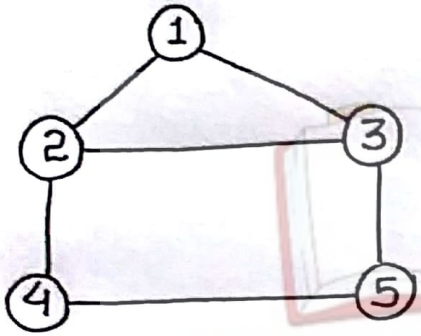


Graph which is a tree

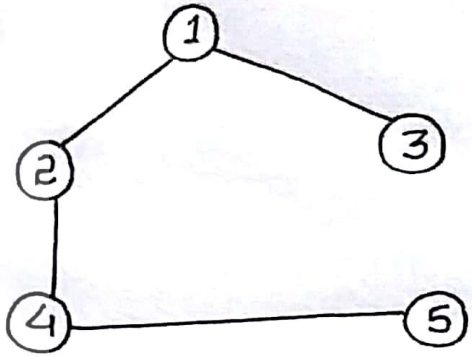
Question:

Which of the following graphs are trees ?

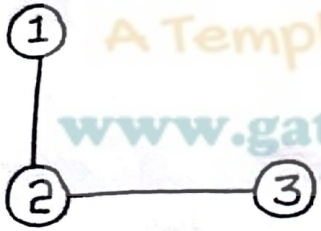
(A)



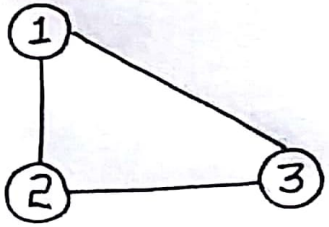
(B)



(C)



(D)



Solution: Only (B) and (C) are trees

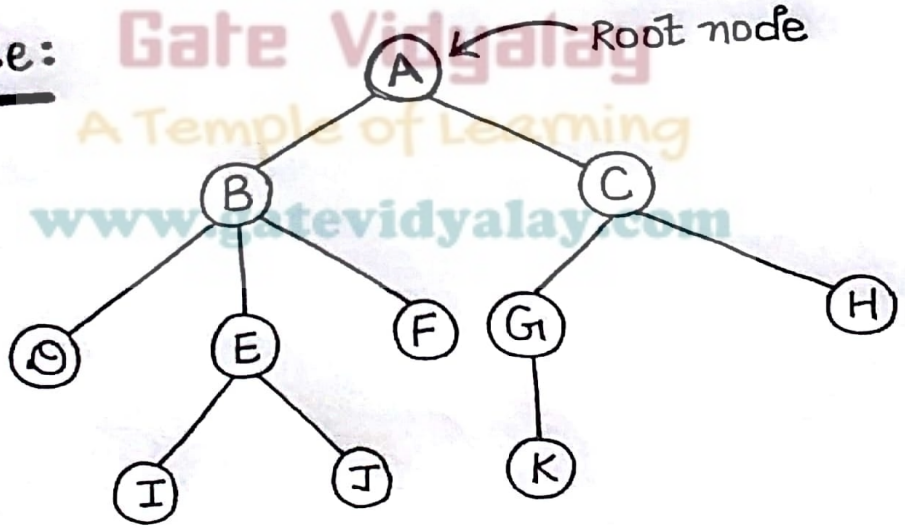
Properties of Trees:

- There is one and only one path between every pair of vertices in a tree.
- A tree with n vertices has $n-1$ edges.
- A graph is a tree if and only if it is minimally connected.
- Any connected graph with n vertices and $n-1$ edges is a tree.

Basic Terminology...:

1) Root:

- Root node is the origin of tree data structure. It is the first node.
- In any tree, there must be only one root node and we can never have multiple root nodes in a tree.
- **Example:**

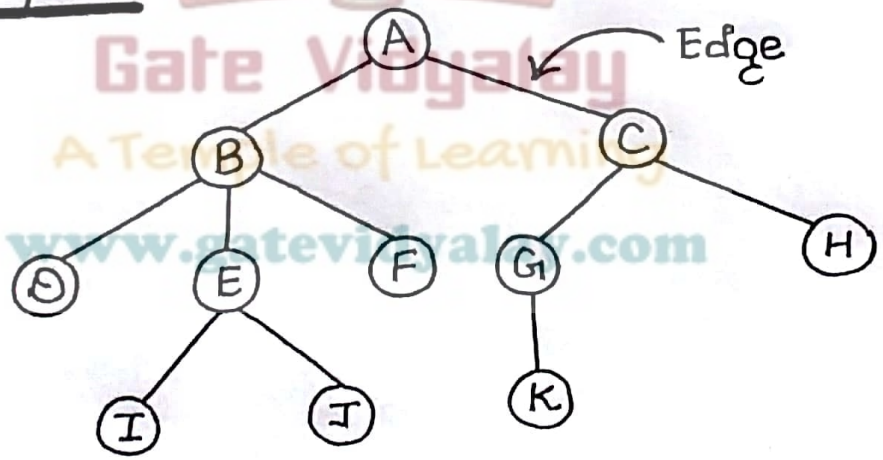


2) Edge:

- The connecting link between any two nodes is called as edge.

Remember: In a tree with 'n' number of nodes, there will be exactly 'n-1' number of edges

- **Example:**

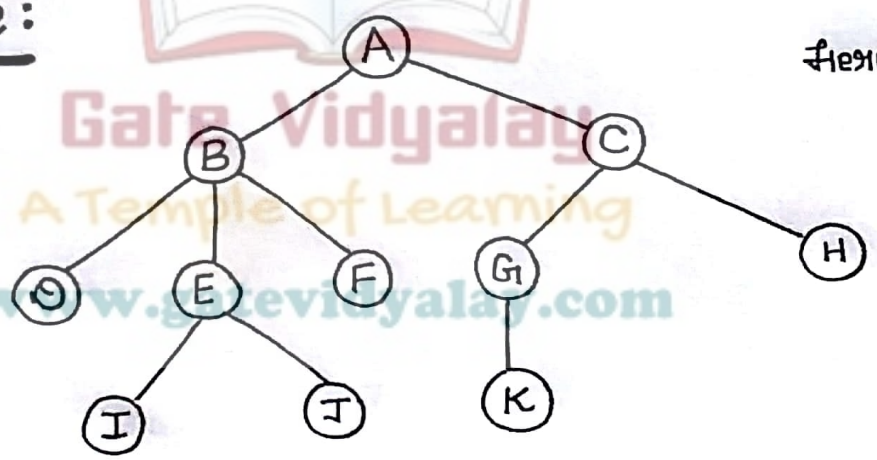


3. Parent:

- The node which has a branch from it to any other node is called as parent node.

  In other words, the node which has child/children is called as parent node.

- **Example:**

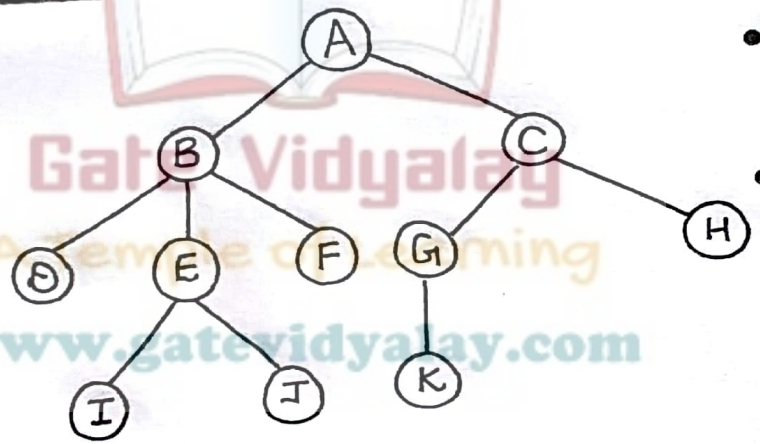


Here, A, B, C, E and G are parent nodes

4. Child :

- The node which is descendant of any node is called as child node . So, all the nodes except root node are child nodes.

- **Example:**



- Here, B and C are children of A
- G and H are children of C
- K is the only child of G etc

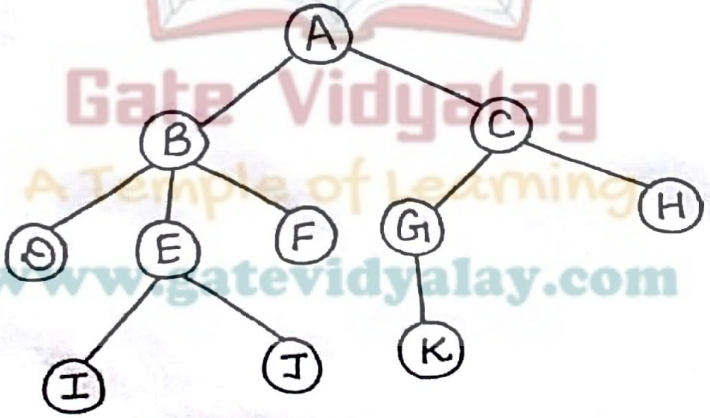
Note: In a tree, any parent node can have any number of child nodes.

5. Siblings :

- Nodes which belong to the same parent are called as siblings.

  In other words, nodes with the same parent are called as sibling nodes.

- **Example :**

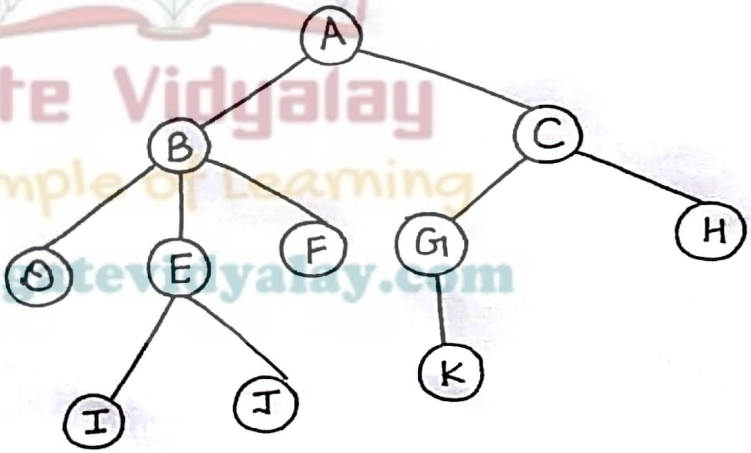


Here,

- B and C are siblings
- D, E and F are siblings
- G and H are siblings
- I and J are siblings

6. Degree:

- The total number of children of a node is called as degree of that node.
- Degree of a tree is the highest degree of a node among all the nodes in the tree.
- **Example:**



Here,

- Degree of B = 3
- Degree of A = 2
- Degree of F = 0

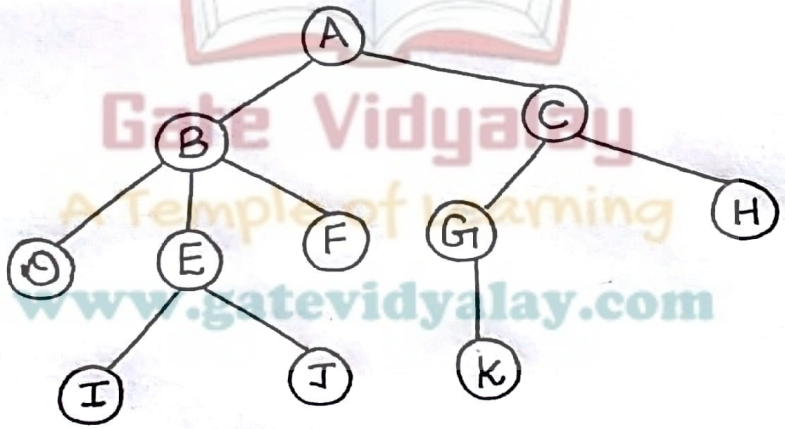
etc

7. Internal Nodes:

- The node which has atleast one child is called as internal node

  They are also called as non-terminal nodes.

- **Example:**



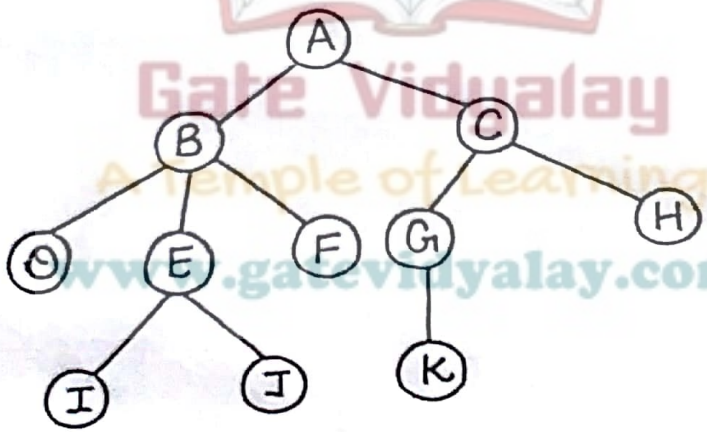
Here,

A, B, C, E and G are internal nodes

Note: Every non-leaf node is an internal node.

8. Leaf Nodes :

- The node which does not have a child is called as leaf node.
- The leaf nodes are also called as external nodes or terminal nodes.
- **Example:**



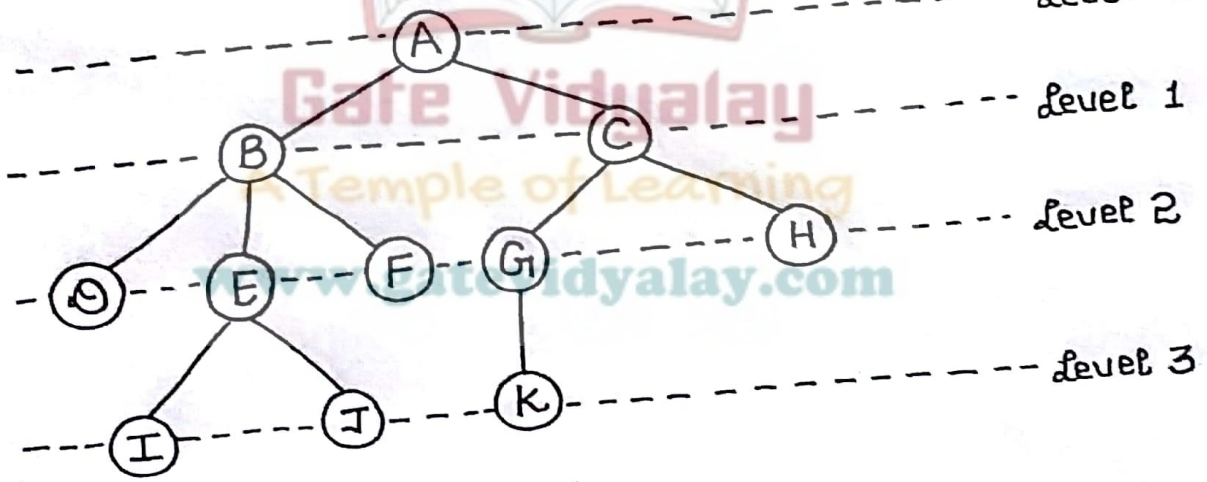
Here,

D, I, J, F, K and H are leaf nodes

9. Level:

- In a tree, each step from top to bottom is called as level and the level count starts with '0' and incremented by one at each level or step.

- **Example:**



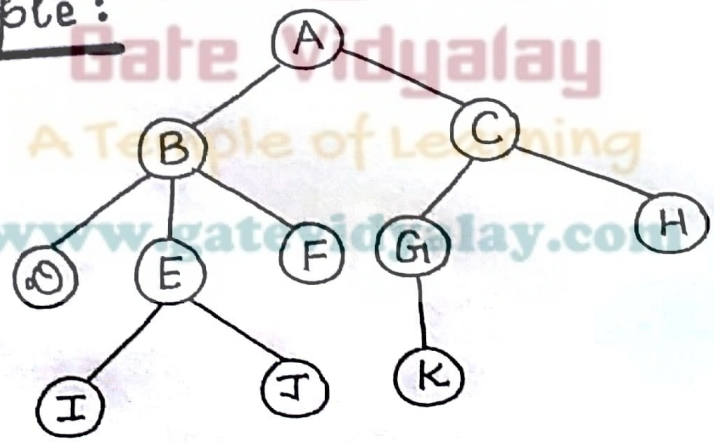
10. Height:

- The total number of edges from leaf node to a particular node in the longest path is called as height of that node.

Note:

- Height of the tree = Height of the root node
- Height of all leaf nodes = 0

- **Example:**



Here,

- Height of K = 0
- Height of B = 2
- Height of A = 3
- Height of G = 1
- Height of H = 0

etc

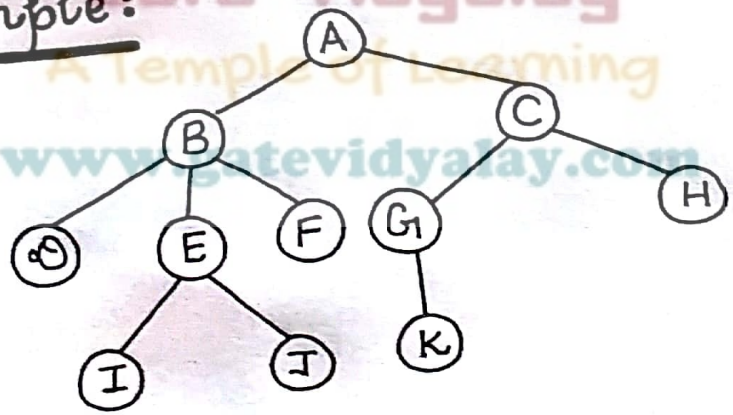
11. Depth:

- The total number of edges from root node to a particular node is called as depth of that node.

Note:

- Depth of the tree = Total number of edges from root node to a leaf node in the longest path
- Depth of the root node = 0

- **Example:**



Here,

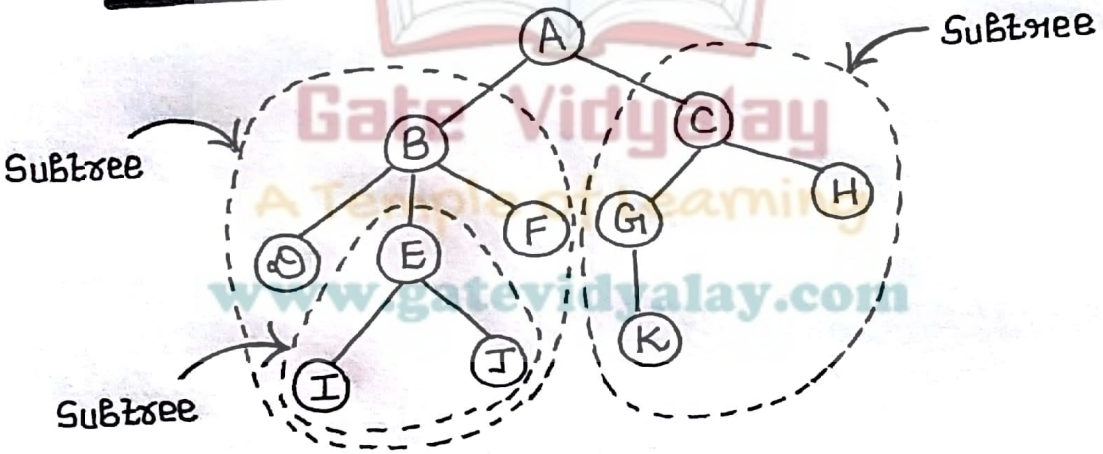
- Depth of A = 0
- Depth of B = 1
- Depth of K = 3

etc.

12. Subtree :

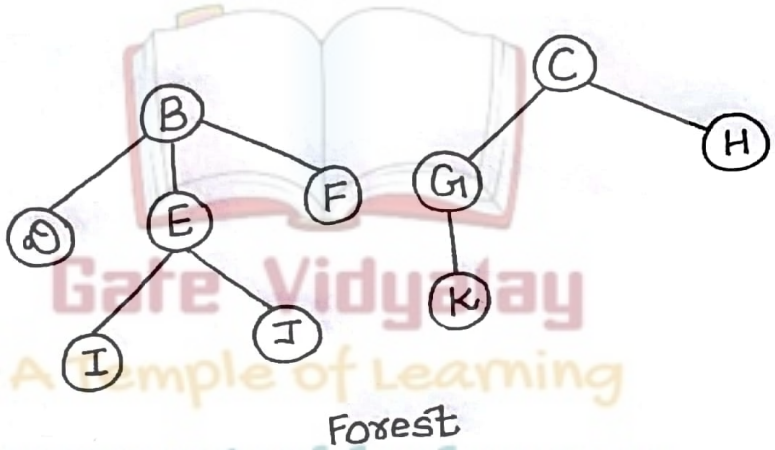
- In a tree, each child from a node forms a subtree recursively. Every child node will form a subtree on its parent node.

- **Example :**



13. Forest:

- It is a set of disjoint trees.
- **Example:**



Forest