Unit 4
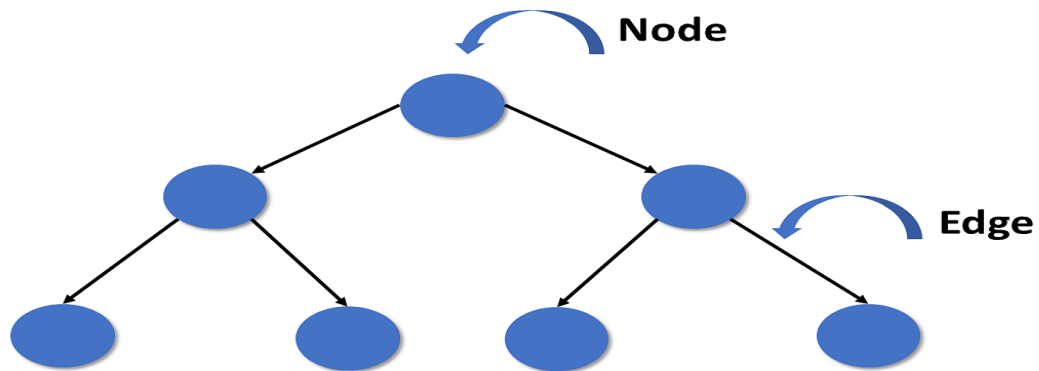**An Introduction to Tree in Data Structure**

The tree is a nonlinear hierarchical data structure and comprises a collection of entities known as nodes. It connects each node in the tree data structure using "edges", both directed and undirected.

.

The Necessity for a Tree in Data Structures

Other data structures like arrays, linked-list, stacks, and queues are linear data structures, and all these data structures store data in sequential order. Time complexity increases with increasing data size to perform operations like insertion and deletion on these linear data structures. But it is not acceptable for today's world of computation.

The non-linear structure of trees enhances the data storing, data accessing, and manipulation processes by employing advanced control methods traversal through it. You will learn about tree traversal in the upcoming section.

Tree Node

A node is a structure that contains a key or value and pointers in its child node in the tree data structure.

In the tree data structure, you can define the tree node as follows.

```
struct node

{

int data;

struct node *leftchild;

struct node *rightchild;

}
```
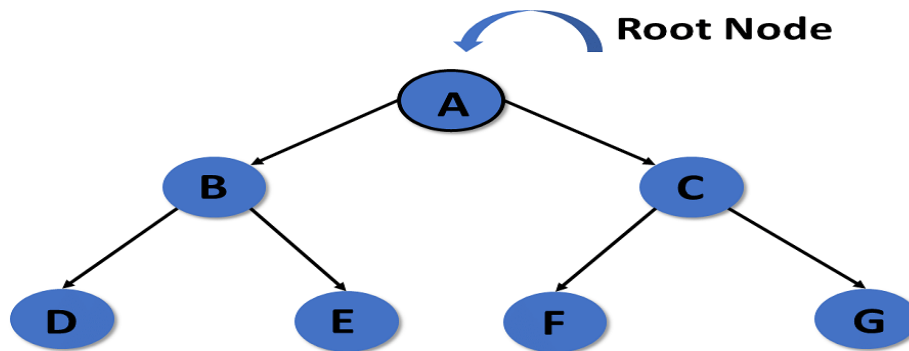
Root Node

| Left Child	Data	Right Child

Tree Terminologies

Root

- In a tree data structure, the root is the first node of the tree. The root node is the initial node of the tree in data structures.

- In the tree data structure, there must be only one root node.


Edge

- In a tree in data structures, the connecting link of any two nodes is called the edge of the tree data structure.

- In the tree data structure, N number of nodes connecting with N -1 number of edges.
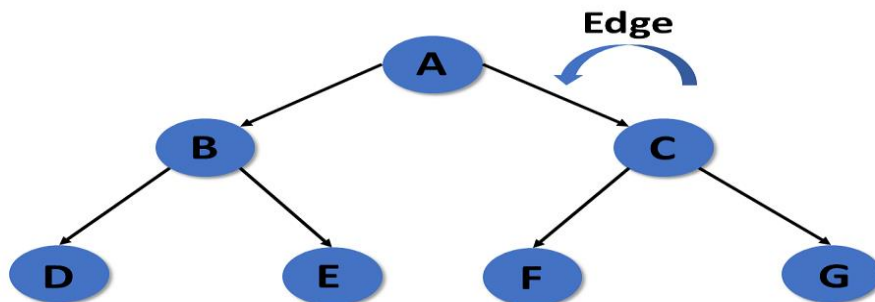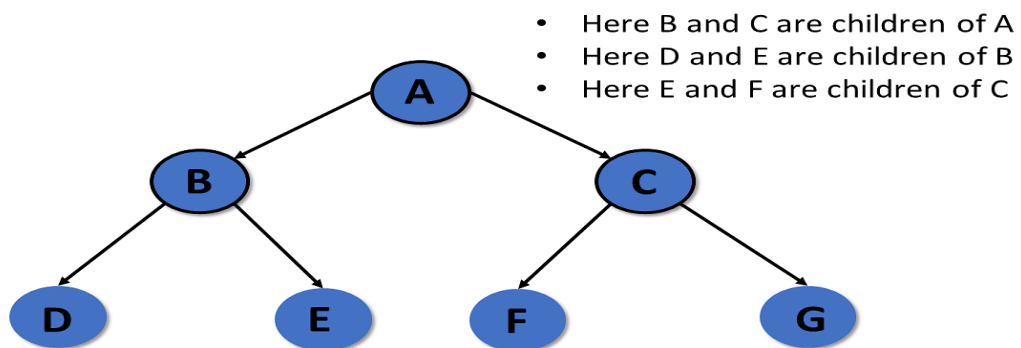
Parent

In the tree in data structures, the node that is the predecessor of any node is known as a parent node, or a node with a branch from itself to any other successive node is called the parent node.

Here A, B and C are parent nodes
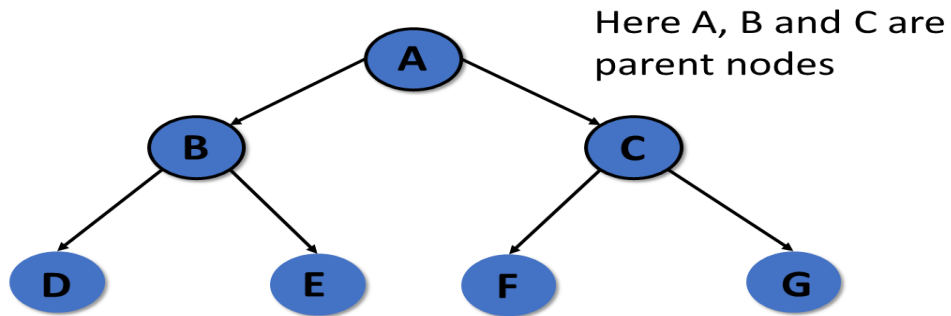
Child

- The node, a descendant of any node, is known as child nodes in data structures.

- In a tree, any number of parent nodes can have any number of child nodes.

- In a tree, every node except the root node is a child node.

- Here B and C are children of A
- Here D and E are children of B
- Here E and F are children of C

Siblings

In trees in the data structure, nodes that belong to the same parent are called siblings.

- Here B and C are siblings
- Here D and E are siblings
- Here F and G are siblings
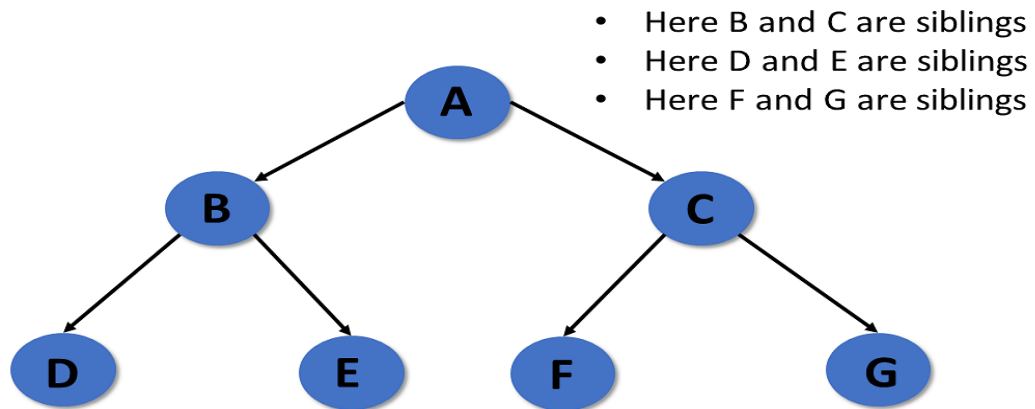
Leaf

- Trees in the data structure, the node with no child, is known as a leaf node.

- In trees, leaf nodes are also called external nodes or terminal nodes.

Here D, E, F and G are leaf nodes.
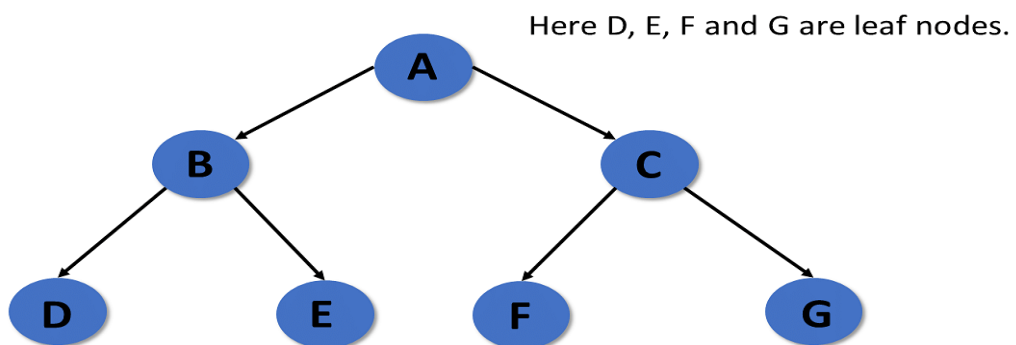
Internal nodes

- Trees in the data structure have at least one child node known as internal nodes.

- In trees, nodes other than leaf nodes are internal nodes.

- Sometimes root nodes are also called internal nodes if the tree has more than one node.

Here A, B and C are internal nodes.
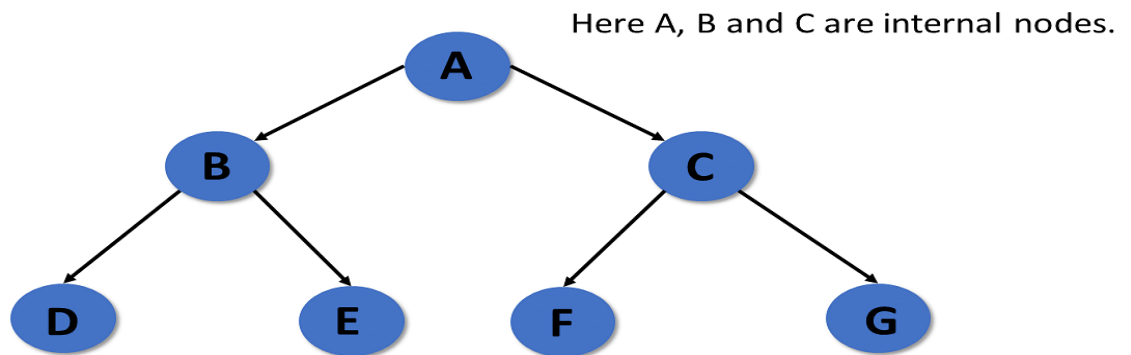
Degree

- In the tree data structure, the total number of children of a node is called the degree of the node.

- The highest degree of the node among all the nodes in a tree is called the Degree of Tree.

- Here degree of A, B and C is 2.
- Here degree of D, E, F and G is 0.
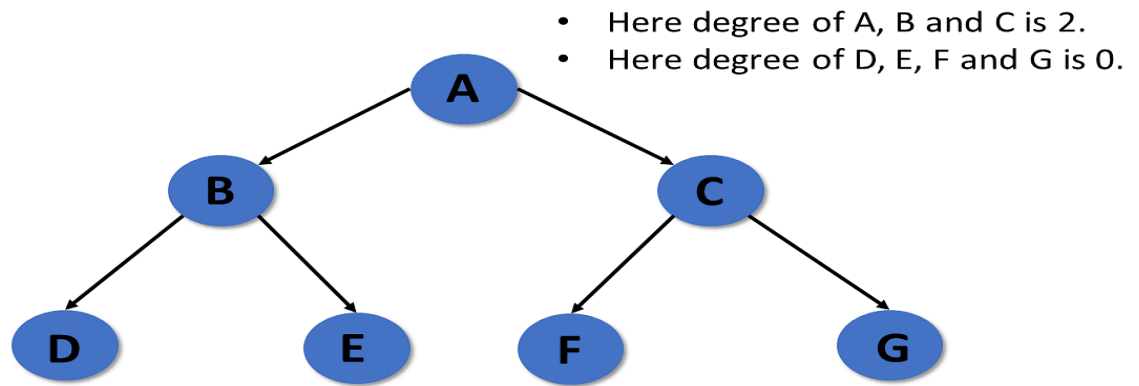
Level

In tree data structures, the root node is said to be at level 0, and the root node's children are at level 1, and the children of that node at level 1 will be level 2, and so on.
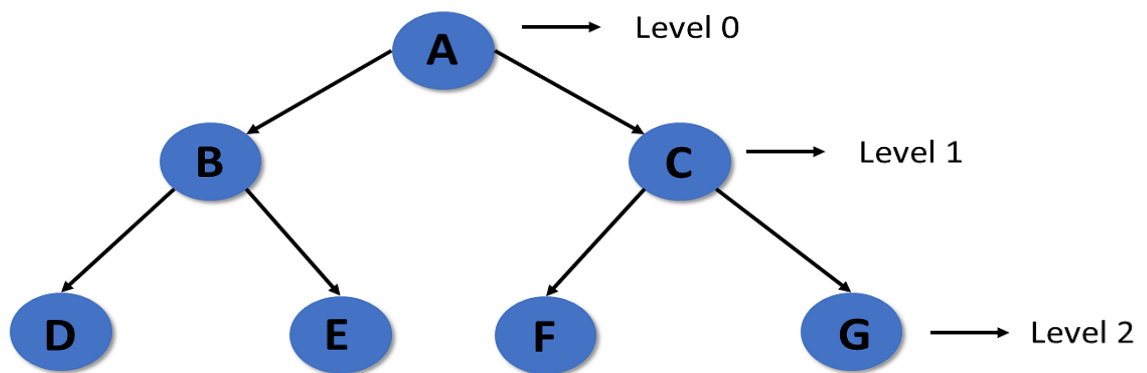
Height

- In a tree data structure, the number of edges from the leaf node to the particular node in the longest path is known as the height of that node.

- In the tree, the height of the root node is called "Height of Tree".

- The tree height of all leaf nodes is 0.
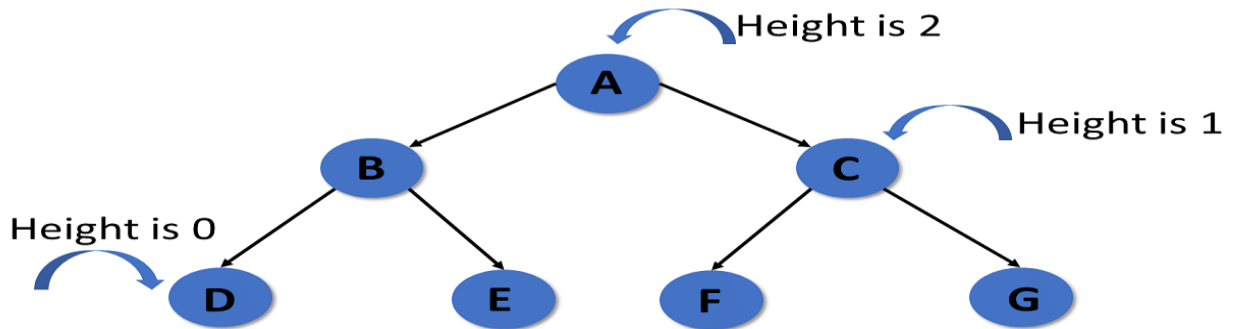
Depth

- In a tree, many edges from the root node to the particular node are called the depth of the tree.

- In the tree, the total number of edges from the root node to the leaf node in the longest path is known as "Depth of Tree".

- In the tree data structures, the depth of the root node is 0.
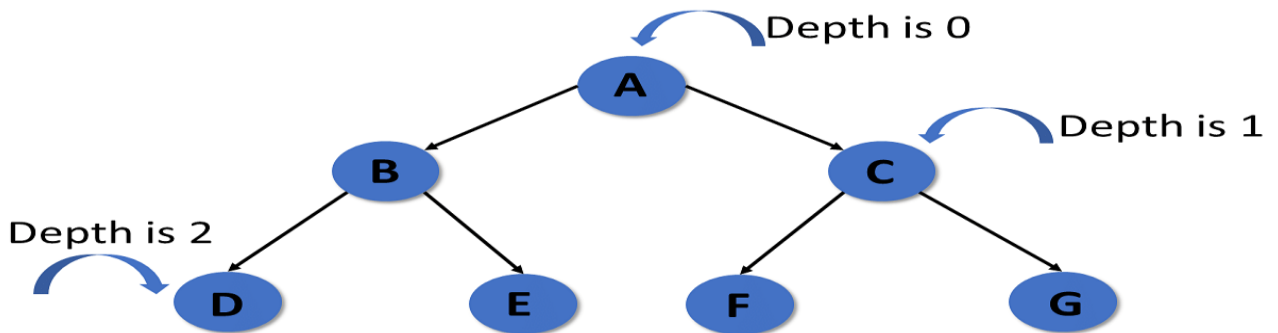
Path

- In the tree in data structures, the sequence of nodes and edges from one node to another node is called the path between those two nodes.

- The length of a path is the total number of nodes in a path.zx

Here the path between A and D is
A-B-D
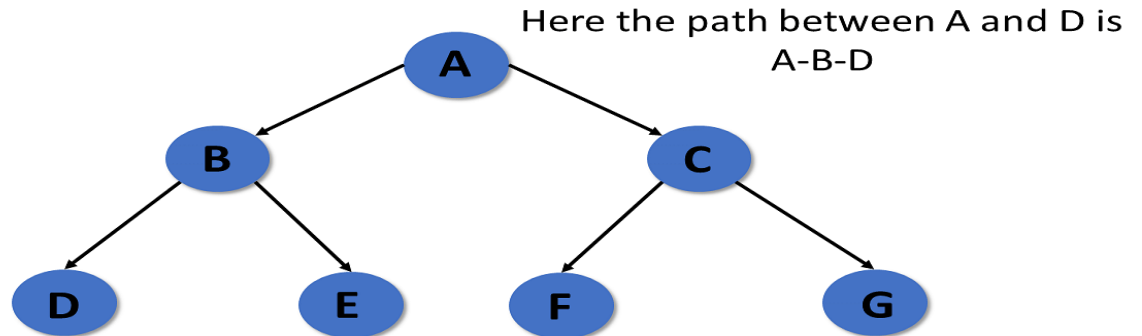
Subtree

In the tree in data structures, each child from a node shapes a sub-tree recursively and every child in the tree will form a sub-tree on its parent node.

Subtree

Subtree