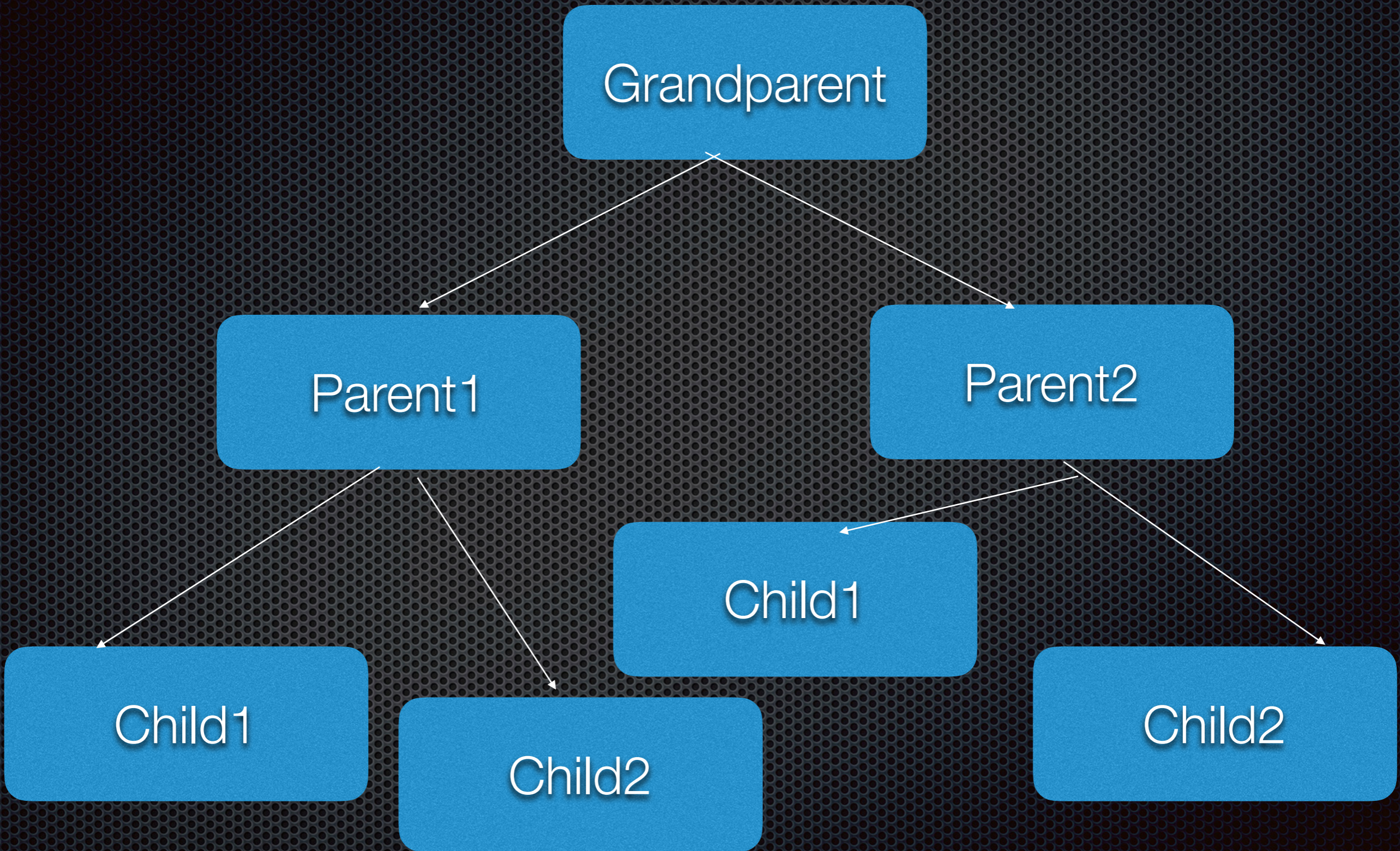


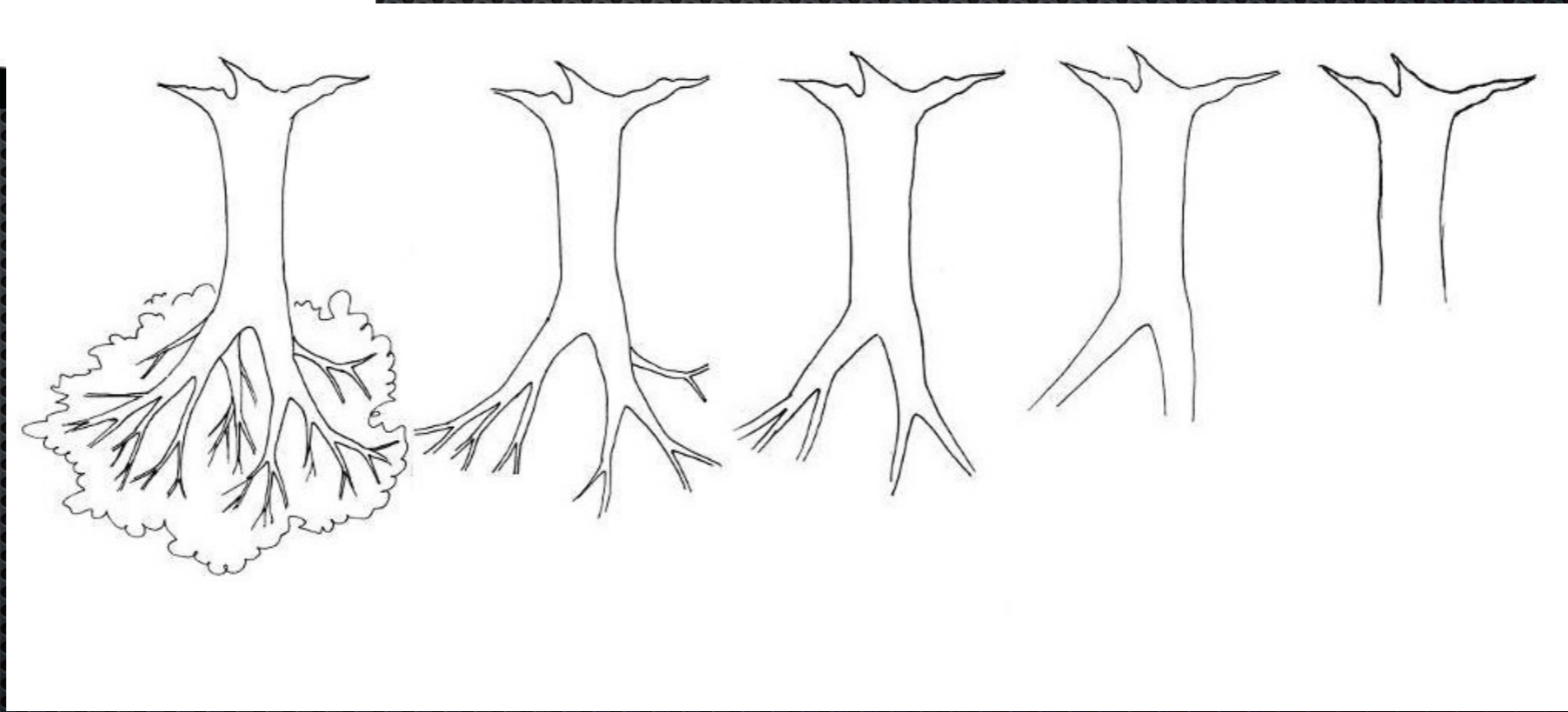
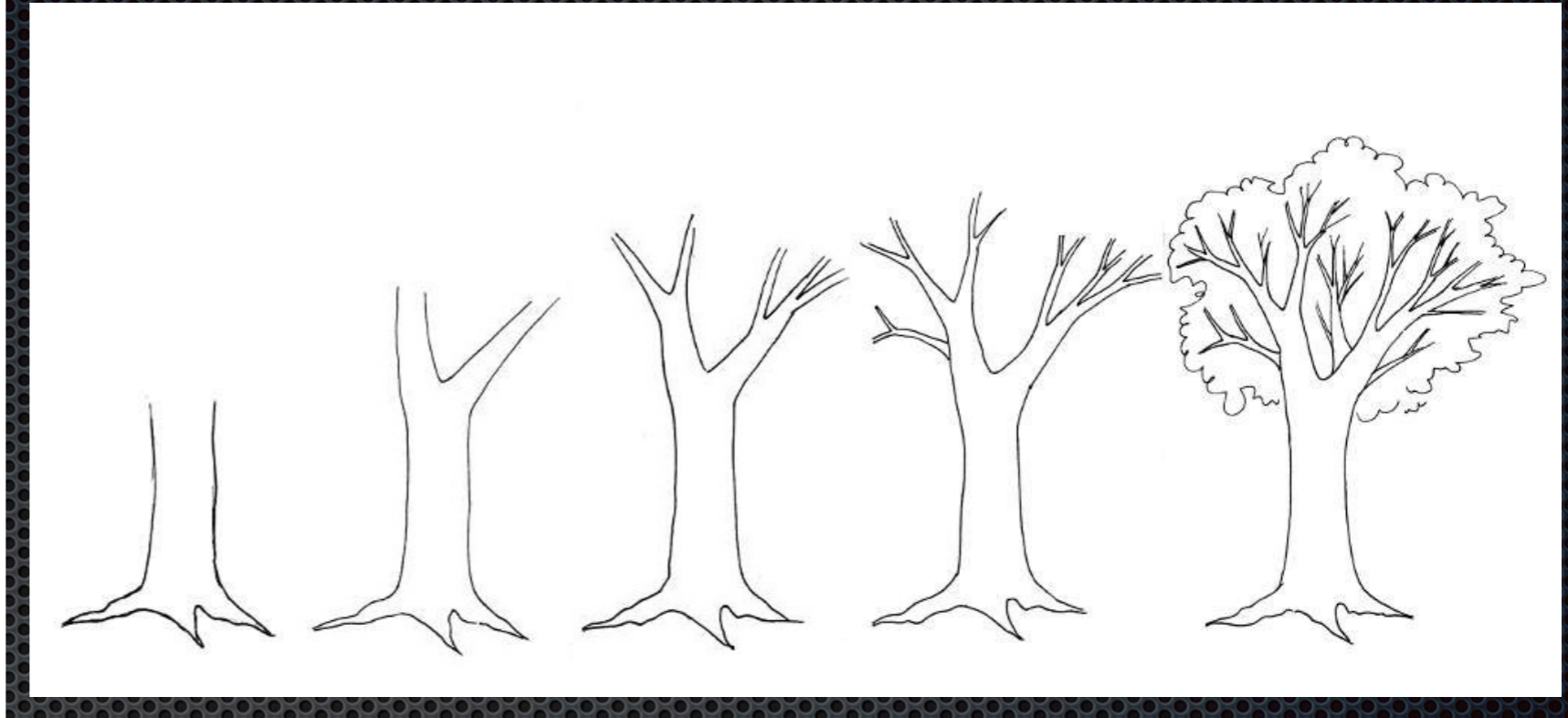
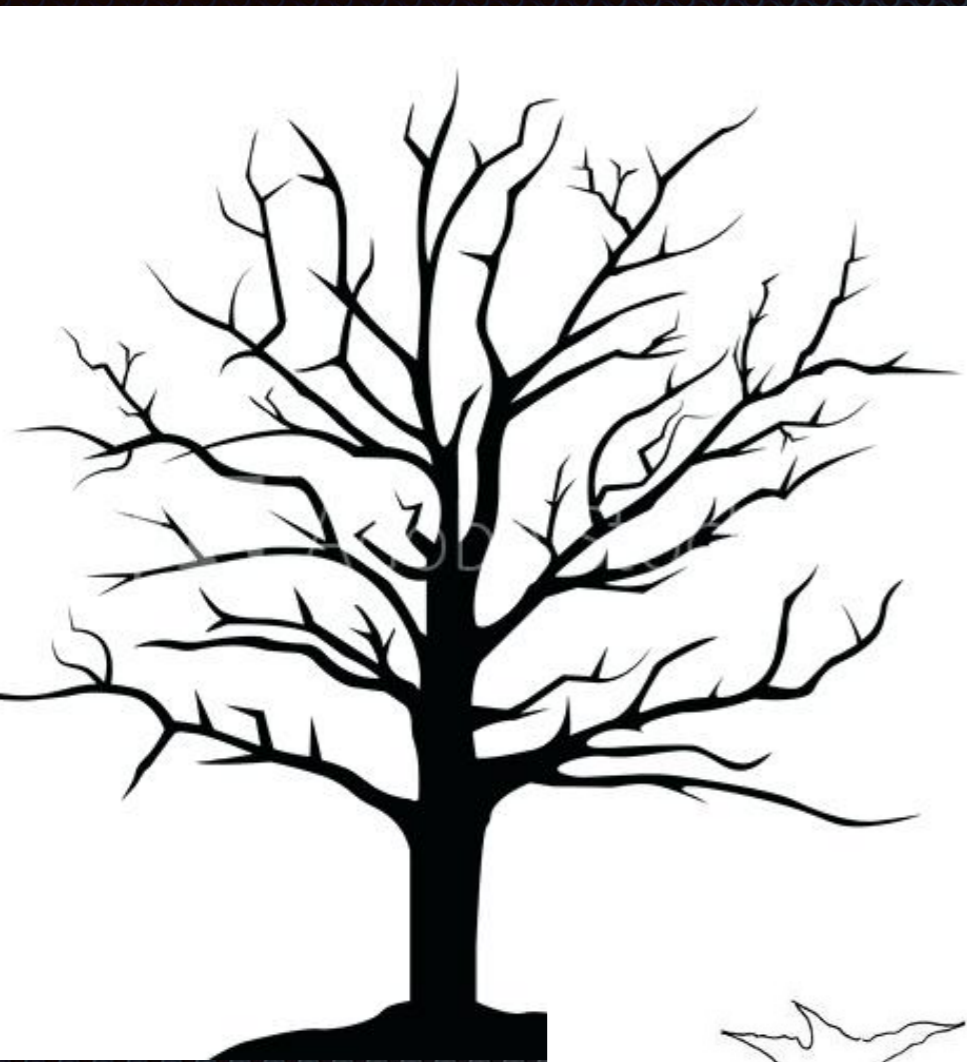
Tree Data Structure

Tree Data Structure

A tree is a nonlinear hierarchical data structure that consists of nodes connected by edges.

it can demonstrate relationships between different nodes with the parent-child hierarchy. So It is also called a hierarchical data structure.





Terminology

Root: the topmost node of the tree

Edge: the link between 2 nodes

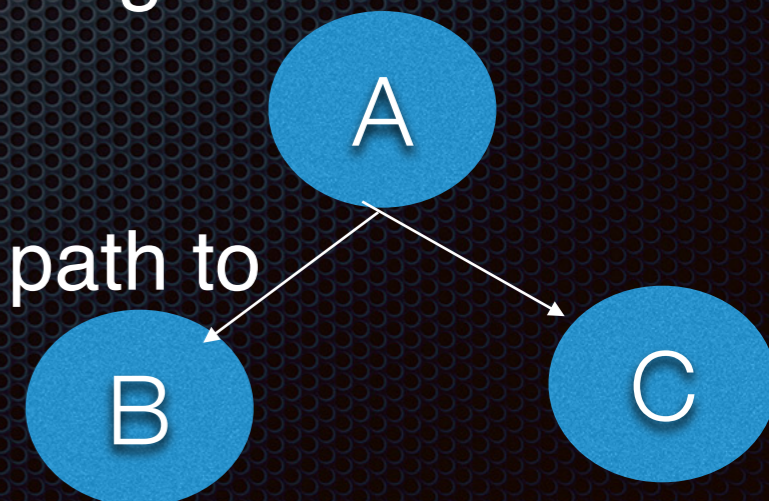
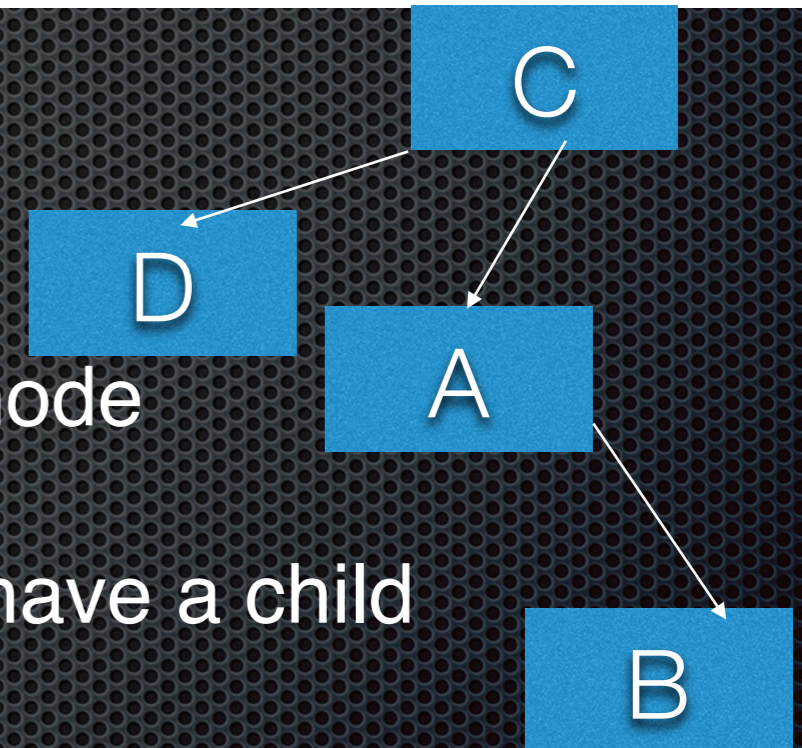
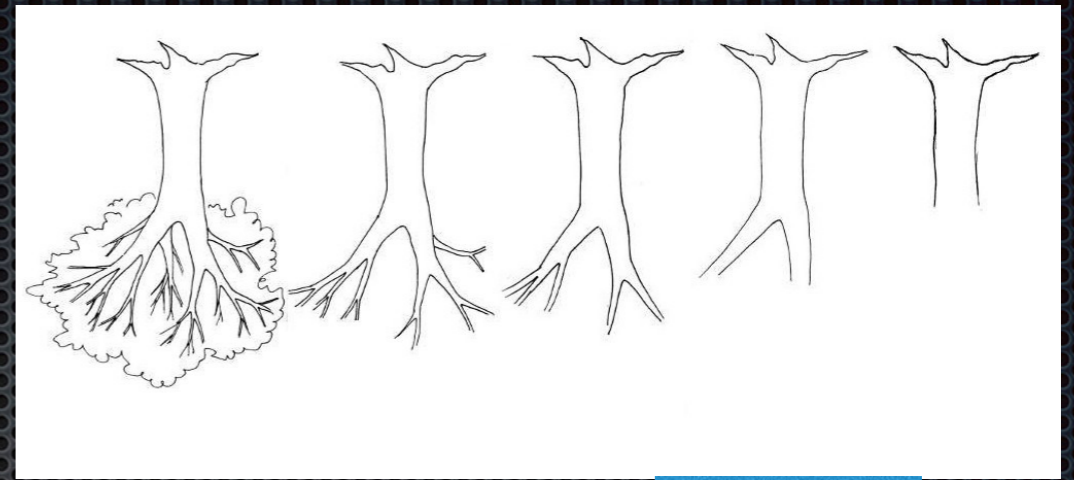
Child: a node that has a parent node

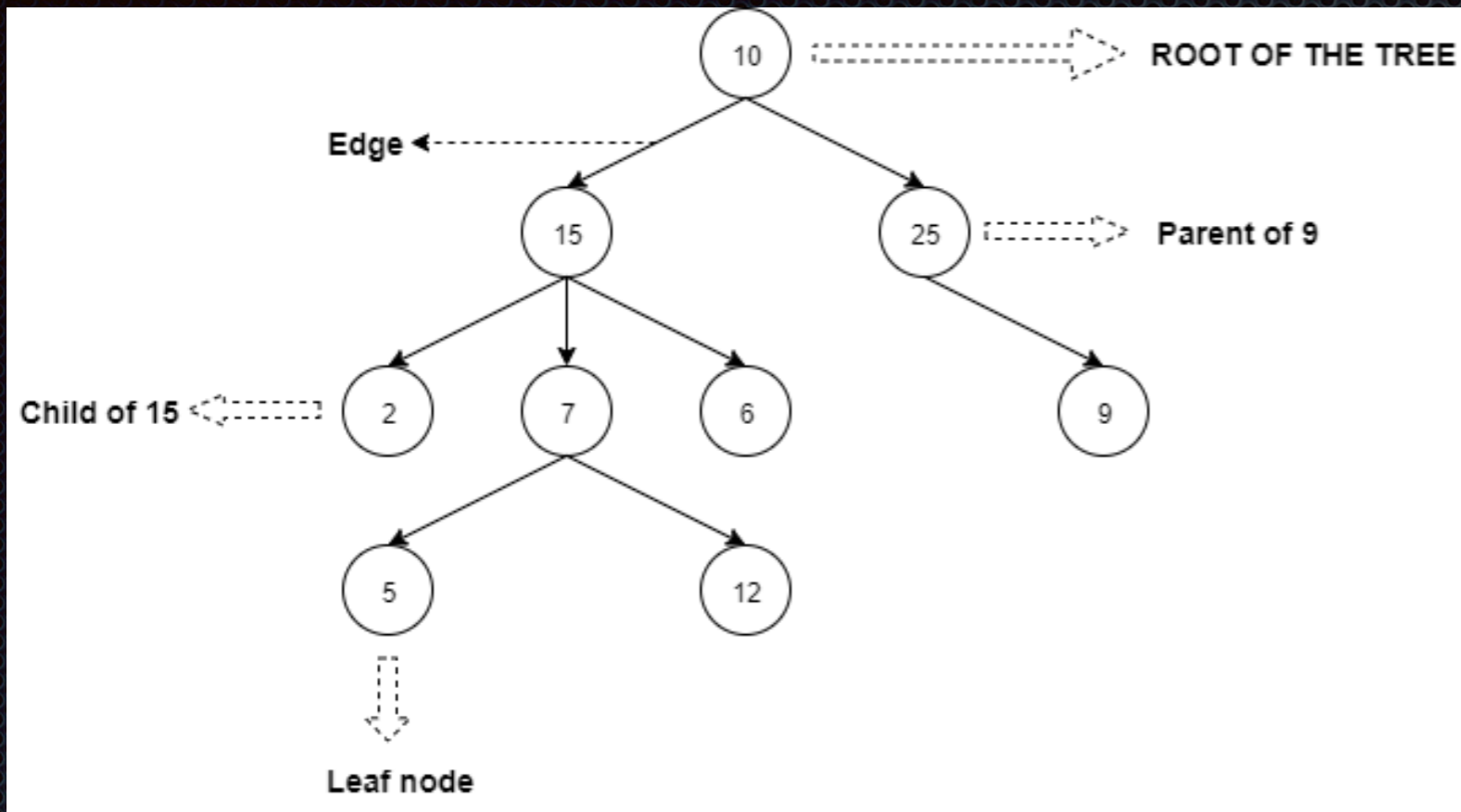
Parent: a node that has an edge to a child node

Leaf(External node) : a node that does not have a child node in the tree

Height: The height of a tree is the length of the longest path to a leaf.

Depth: The depth of a node is the length of the path to its root.





Depth 0 Level 0

Depth 1 Level 1

Depth 2 Level 2

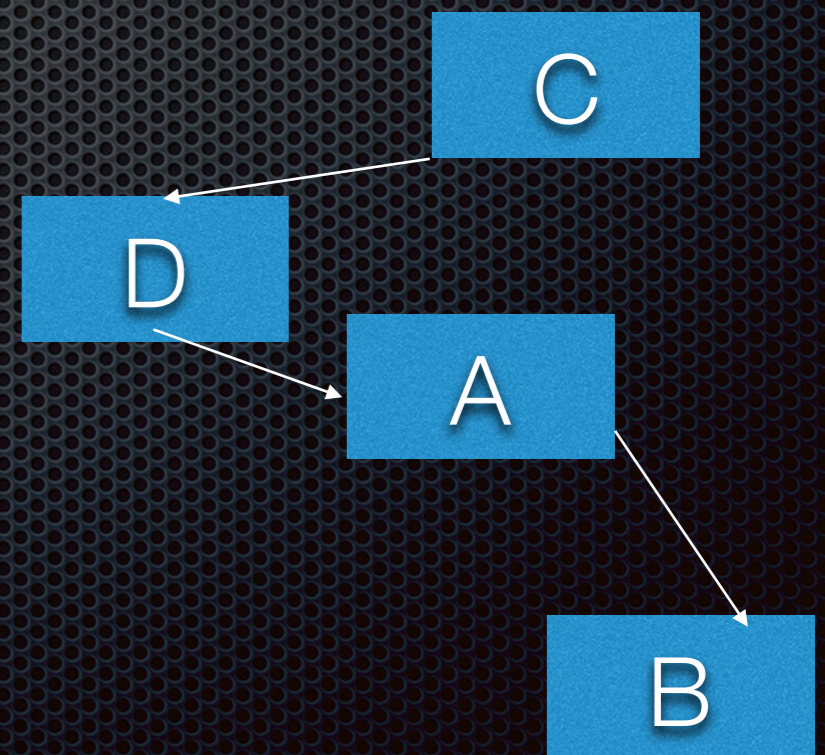
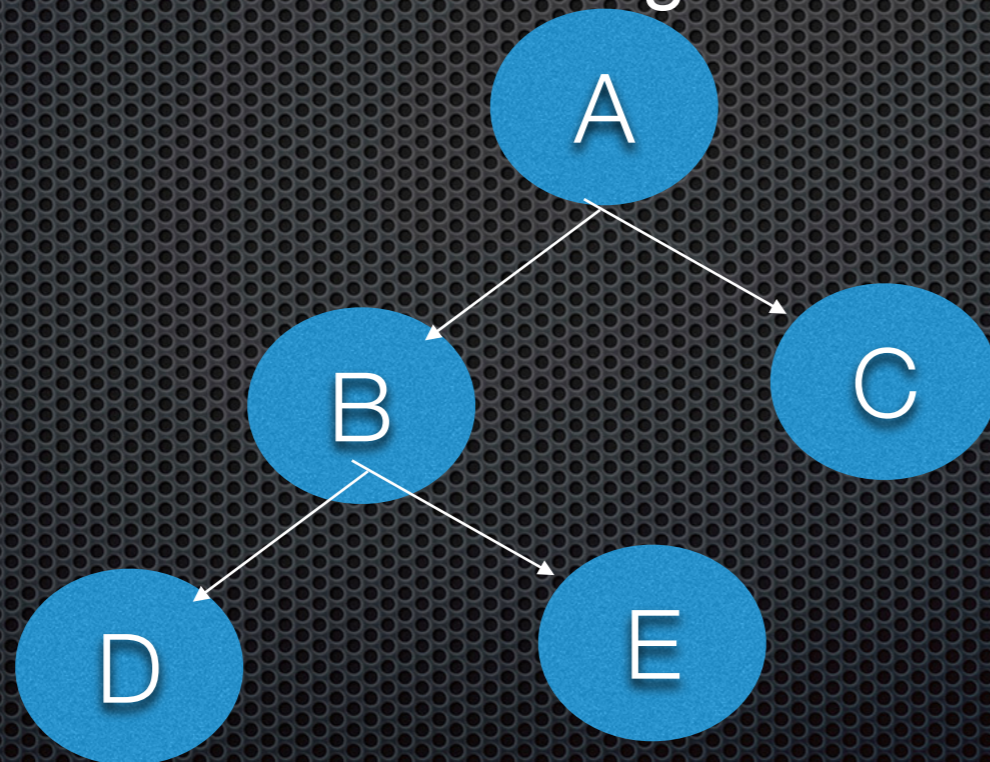
Depth 3 Level 3

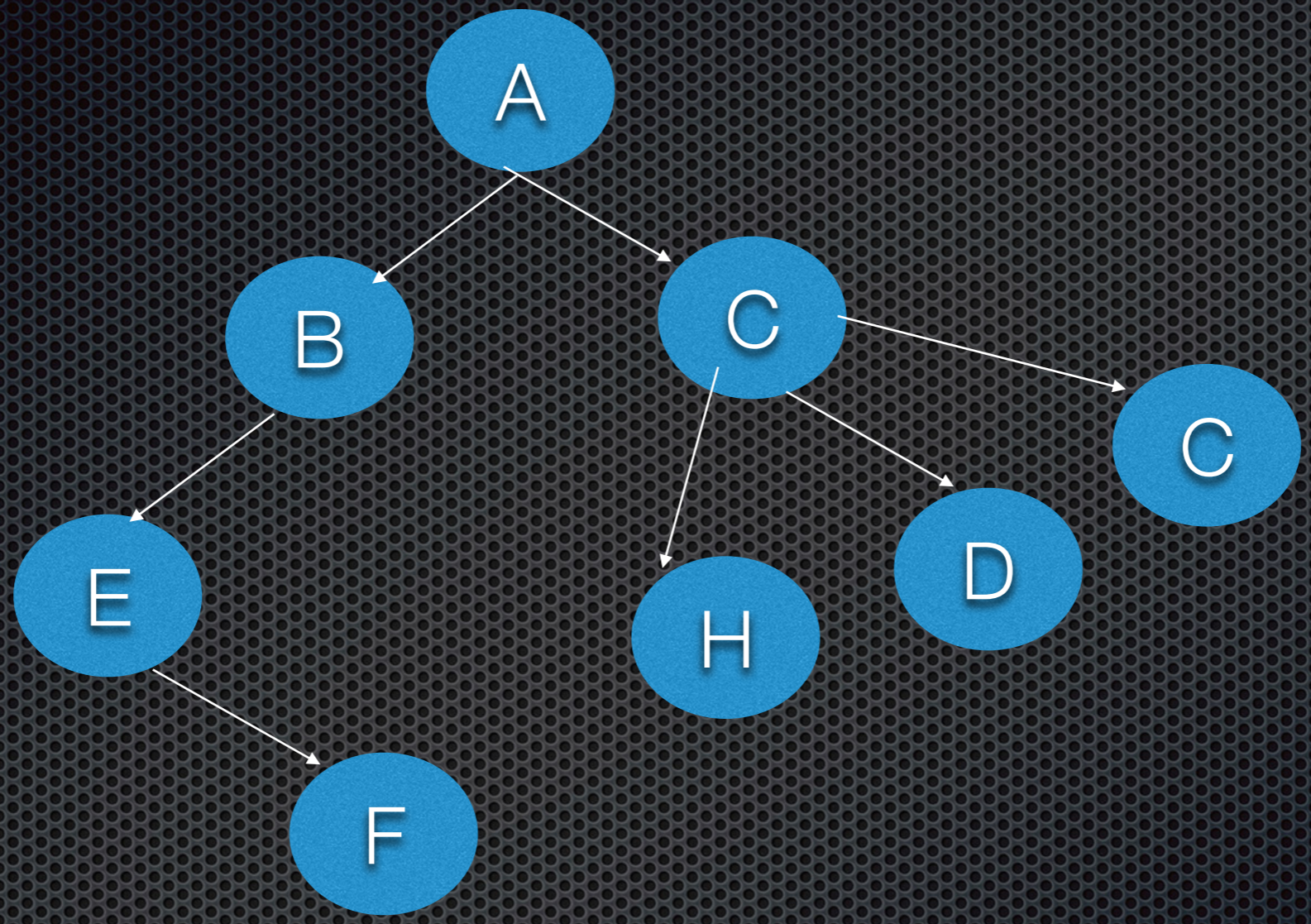
Path: sequence of Nodes and Edges from one node to another node is called as PATH between that two Nodes

Levels : Level of a node represents the generation of a node. If the root node is at level 0, then its next child node is at level 1, its grandchild is at level 2, and so on.

Subtree : A subtree is a portion of a tree data structure that can be viewed as a complete tree in itself.

Siblings : Nodes which belong to the same parent are called as siblings



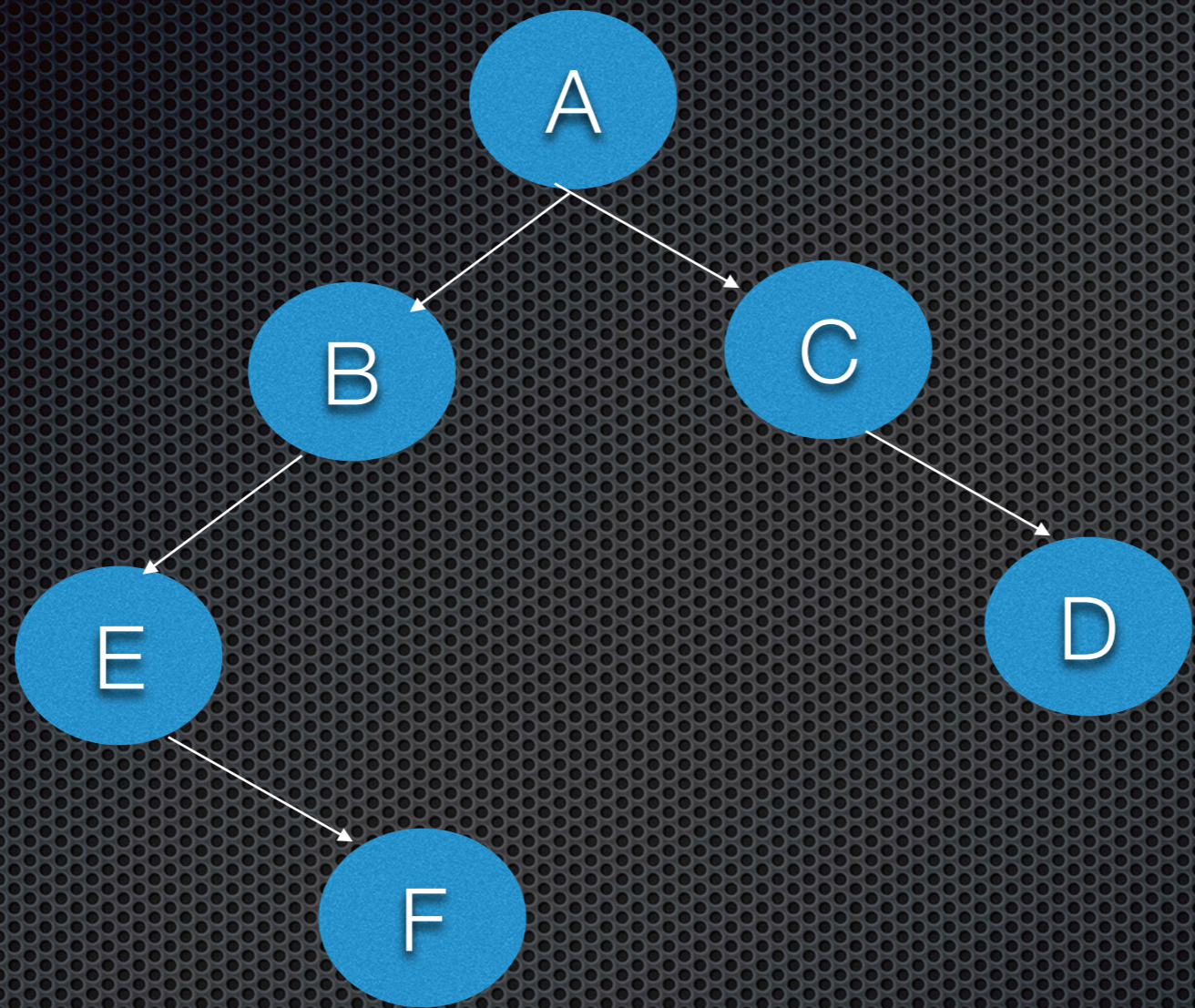


Degree :Degree of a node is the total number of children of that node.

Degree of a tree is the highest degree of a node among all the nodes in the tree.

Terminal node : A node with degree zero is call a terminal node or a leaf.

Non-terminal or internal node : Any node (except the root node) whose degree is not zero is called non-terminal node. (node which has at least one child)



A tree is a collection of entities called nodes. Nodes are connected by edges. Each node contains a value or data, and it may or may not have a child node .

Node : node is a structure which may contain a value, address of Child node

```
Struct node{  
    Int val;  
    Struct node *left;  
    Struct node *right;  
}v1;  
V1.val=55;
```

