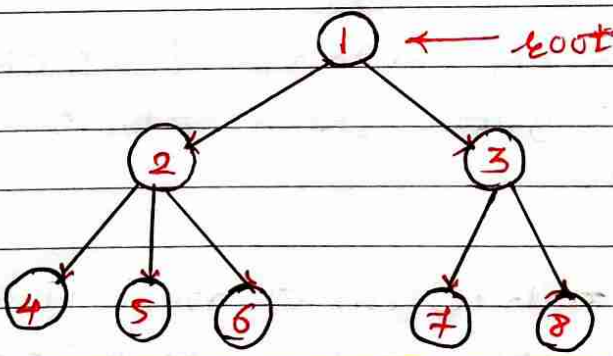
Tree :- We read data structure, like an array, linked list, stack and queue in which all elements are arranged in a sequential manner.

A tree is one of the data structures that represents hierarchical data.

defination :- A tree is a data structure defined as collection of objects or entities known as nodes that are linked together to represent or simulate hierarchy.

- A tree is a non-linear data structure because it does not store in a sequential manner. It is a hierarchical structure as elements in tree are arranged in multiple levels.
- In tree data structure topmost node is called as root node. Each node contains some data & data can be of any type.
- Each node contains some data & links or reference of other nodes that can be called children.



Some Basic terms of tree :-

1). **link :-** Each node is labeled with some number. each array shown in fig is known as link between two nodes.

2). **Root :-** The root node is top most node in tree hierarchy. root node is one that doesn't have any parent. If node is directly linked to some other

node, then it would be called a parent-child relation-ship.

3). child node :- If the node is a descendant of any node, then node is called as child node.

4). parent :- If node contains any sub-node, then node is said to be parent of that sub-node.

5). sibling :- The nodes that have same parents are called siblings.

6). leaf node :- node which doesn't have any child node, a leaf a bottom-most node of tree.

7). ancestor node :- It is any predessor node on a path from root to that node. In the given fig, 1, 2, 5 are ancestors of node 10.

8). Descendant :- The immediate successor of given node is known as descendant of a node.

* Properties of tree data structures :-

1). Recursive data structure :- Tree is also known as recursive data structure. Recursion means reducing something in a self-similar manner.

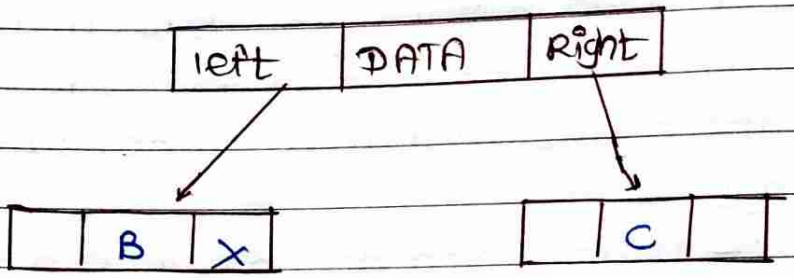
2). Number of edges :- If there are (n) nodes, then there would be (n-1) edges. each node, except root node, will have atleast one incoming line known as an edge.

3). Depth of node x :- It can be defined as length of path from root to node x. one edge contributes one unit length in the path, depth can be defined as no. of edges between root node and node (x)
The root node has depth 0.

4). Height of node x :- It is defined as longest path from node x to leaf node.

Implementation of tree :-

The tree data structure can be created nodes dynamically with help of pointers. The tree in memory can be represented as shown :




```
Struct node
 {
  int data;
 struct node *left;
 struct node *right;
 }
```

The above structure can only be defined for binary trees because binary tree can have utmost two children, and generic trees.

Application of trees :-

1). storing naturally hierarchical data :- File system., stored on disc drive, file and folder are in form of naturally heirarchical data and store in form of trees.

2). organize data :- It is used to organize data for efficient insertion, deletion and searching.

3). Trie :- It is special kind of tree that is used to store dictionary. It is fast and efficient way for dynamic spell checking.

4). Heap :- It is also a tree data structure implemented using arrays. It is used to implement priority queues.