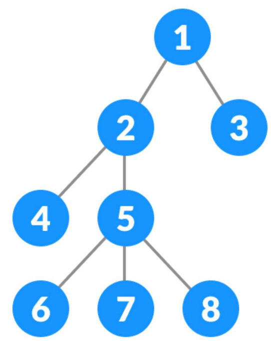
Lecture

NOTE: FOR FURTHER DETAILS AND MORE COMPREHENSIVE STUDY, PLEASE SEE RECOMMENDED BOOKS OR INTERNET.

Trees

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



Why Tree?

Other data structures such as arrays, linked list, stack, and queue are ***linear*** data structures that store data sequentially. In order to perform any operation in a linear data structure, the time complexity increases with the increase in the data size. But, it is not acceptable in today's computational world. Consider a search operation in a linear or sequential data structure **what will be the cost of search operation?**
Different tree data structures allow quicker and easier access to the data as it is a non-linear data structure.

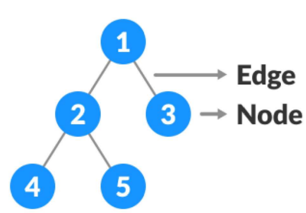
Tree Terminologies

Node

A ***node*** is an entity that contains a key or value and pointers to its child nodes. The last nodes of each path are called ***leaf*** nodes that do not contain any child nodes. The node having at least a child node is called an ***internal*** node.

Edge

It is the ***link*** between any two nodes.



Root
It is the ***topmost node*** of a tree from which the tree starts. There is only one root per tree and one path from the root node to any node.

Parent
Any node except the root node has one ***edge upward*** to a node called ***parent***.

Child
The node below a given node connected by its ***edge downward*** is called its ***child*** node.

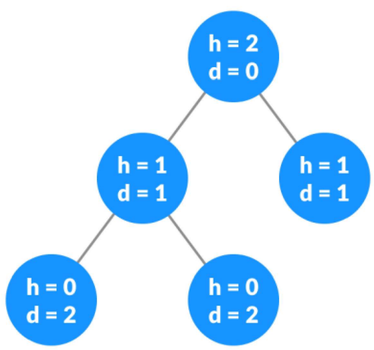
Leaf
The node which ***does not have any child node*** is called the ***leaf*** node.

Height of a Node
The height of a node is the number of ***edges*** from the ***node to the deepest leaf***.

Depth of a Node
The depth of a node is the number of ***edges*** from the ***root to the node***.

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.

Height of a Tree
The height of a Tree is the ***height of the root node*** or the ***depth of the deepest node***.



Degree of a Node
The degree of a node is the total ***number of children*** of that ***node***.

Visiting
Visiting refers to ***checking the value*** of a ***node*** when control is on the node.

Tree Traversal

In order to perform any operation on a tree, you need to reach to the ***specific node***. The tree traversal algorithm helps in ***visiting*** a required node in the tree. Traversal means passing through the nodes of a tree in a specific order.

Keys

Key represents a value of a node based on which a search operation is to be carried out for a node.

Path

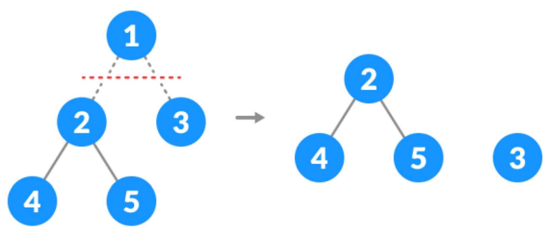
Path refers to the ***sequence of nodes*** along the edges of a tree.

Subtree

Subtree represents the ***descendants*** of a ***node***.

Forest

A collection of ***disjoint trees*** is called a forest. You can create a forest by cutting the root of a tree.



Types of Tree

1. ***Binary Tree***
2. ***Binary Search Tree***
3. ***AVL Tree***
4. ***B-Tree***

Representation of a Tree in memory

Tree is represented as a linked list with two references to a child node and sometimes a reference to a parent node.

```
class Node <T>
{
    T data;
    Node LeftChild;
    Node RightChild;
}
```

Applications of Trees

1. Binary Search Trees are used to quickly check whether an element is present in a set or not.
2. Trees are used to create the data compression coding.
3. Trees are used for Expression solution.
4. Trees are used for decision making algorithms.
5. Heap is a kind of tree that is used for heap sort.
6. A modified version of a tree called Tries is used in modern routers to store routing information.
7. Most popular databases use B-Trees and T-Trees, which are variants of the tree structure we learned above to store their data
8. Compilers use a syntax tree to validate the syntax of every program you write.