**Tree:**

A tree (T) is a non-linear data structure which consist of finite non-empty set of elements. One of the elements is called the Root node and the remaining elements (if any) are divided into sub-trees.

**Binary Tree:**

A binary tree (T) is either empty or has a finite number of elements.

When the binary tree is non-empty, one of the elements is called the root and the remaining elements (if any) can be partitioned into at most two sub-trees, called The Left and Right sub-trees of T.

**Degree of an Element:**

The number of child nodes it has deg(tree): MAX value among the deg of its nodes.

**Q. For any non-empty Binary Tree(T), if n0 be the number of terminal nodes (leaf node)**

**and n2 be the number of nodes of deg(2). Then prove that n0=n2+1.**

Let n1 be the number of nodes of deg(1) and n be the total number of nodes.

Since, all the nodes in T are of deg(<=2),

we can write:

n=n0+n1+n2........(i)

Now, if we count the number of edges in a binary tree, we see that every node except the root node

has an edge leading into it. If E is the total number of edges in T, then we can write:

n = E+1..............(ii)

All the edges either eminate from a node of deg(1) or from node deg(2).

Hence, we can write E = n1+2(n2).......(iii)

Therefore, from equation ii and iii, we can write n = 1+n1+2n2.......(iv)

Hence, from equation i and iv we have n0+n1+n2 = 1+n1+2n2.

**Full and Complete Binary Tree**

**Full Binary Tree:**

A full binary tree is a tree in which all the leaf nodes are on the same level. And every non leaf node has exactly two child nodes.

**Complete Binary Tree:**

A complete binary tree is a tree that is either full or full-through next to last level, with the leaf nodes on the last level as far to the left as possible.

**Binary Search Tree:**

A binary search tree (T) is a binary tree that is either empty or if its non-empty it should satisfy the following conditions –

1. The value in the left sub-tree of the root node should be smaller than the value of its parent node.

2. The value of the right sub-tree of the root node should be greater than the value of its parent node.

**Q. Create a Binary Search Tree with the following elements –**

45, 65, 30, 40, 15, 75, 42.

**Traversal in a Binary Search Tree:**

In binary search tree we can traverse in three different ways –

1. Pre-Order

Rules:

Root, Left, Right – 45, 30, 15, 40, 42, 65, 75.

2. In-Order

Rules:

Left, Root, Right – 15, 30, 40, 42, 45, 65, 75.

3. Post-Order

Rules:

Left, Right, Root – 15, 42, 40, 30, 75, 65, 45.

**Construct a Binary Search Tree from the followings –**

In-Order – B, D, A, E, H, G, I, F, C

Post-Order – D, B, H, I, G, F, E, C, A

**Construct a Binary Search Tree with the following –**

In-Order – 12, 45, 50, 51, 52, 72, 83.

Pre-Order – 52, 45, 12, 50, 51, 72, 83.