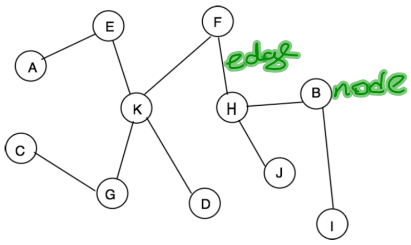
Trees

A **tree** consists of a set of nodes and a set of edges connecting pairs of nodes. A tree has the property that there is exactly one path (no more, no less) between any pair of nodes.

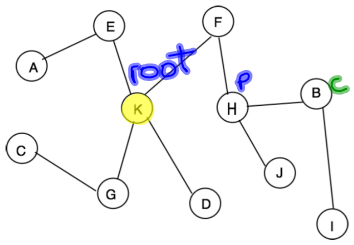


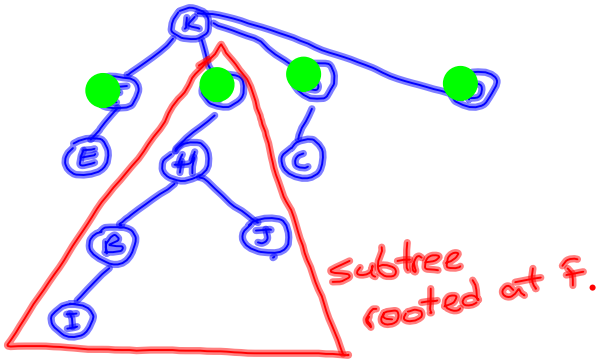
Tree is a graph without cycle

A **path** is a connected sequence of edges. A-E-K-D

Rooted Tree

In a **rooted tree,** one distinguished node is called the root.





Every node c, except the root, has exactly one parent node p, which is the **first node** after c on the path from c to the root. Node c is node p child.

The root has no parent. A node can have any number of children.

A **leaf** is a node with no children.

Siblings are nodes with the same parent.

The **ancestors** of a node d are the nodes on the path from d to the root. This includes node d.

If **node a** is an ancestor of node d, then d is a **descendant** of a.

The **length** of a path is the number of edges in the path.

The **depth** of a node n is the length of the path from n to the root. (The depth of the root is zero.)

depth(F) = 1

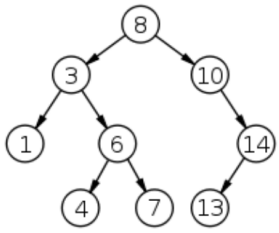
The height of a node n is the length of the path from n to its deepest descendant. (The height of a leaf node is zero.)

height(F) = 3

The height of a tree is the depth of its deepest node = height of the root.

The subtree rooted at node n is the tree formed by n and its descendants. What is the subtree rooted at node F?

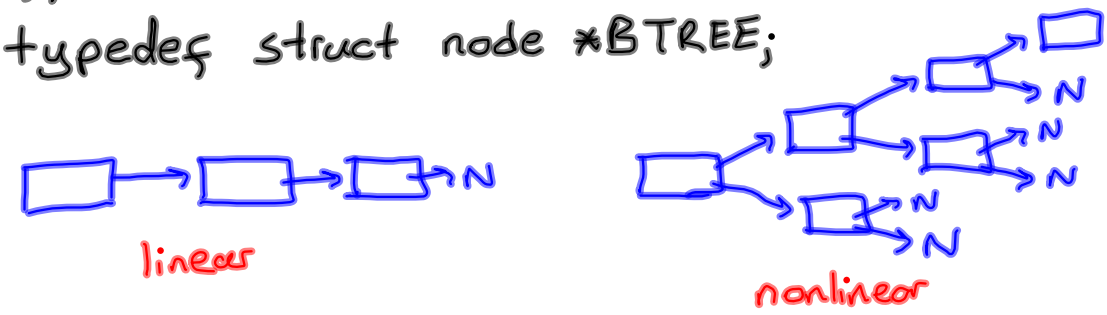
Binary tree



A **binary tree** is a tree in which no node has more than two children, and every child is either a left child or a right child even if it is the only child its parent has.

Data Structure of BTs

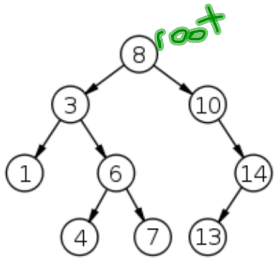
```
struct node {
        int data;
        struct node *left;
        struct node *right;
};
typedef struct node *BTREE;
```



Tree Traversals

A manner of visiting each node is a tree once.

Preorder Traversal: Root - Left - Right.
Inorder " : Left - Root - Right.
Postorder " : Left - Right - Root.



BST

Preorder: 8, 3, 1, 6, 4, 7, 10, 14, 13
Inorder : 1, 3, 4, 6, 7, 8, 10, 13, 14
Postorder : 1, 4, 7, 6, 3, 13, 14, 10, 8