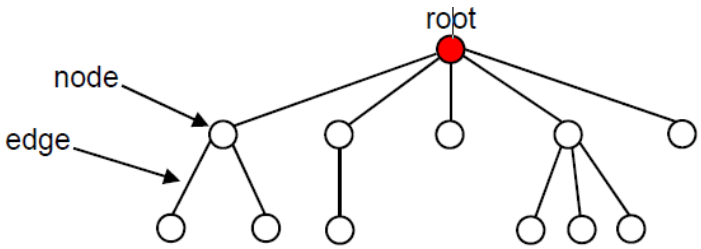
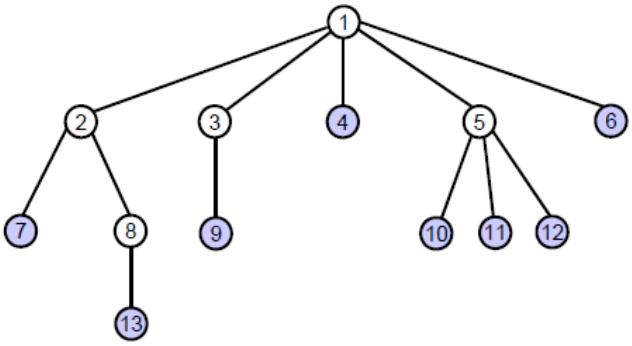
Trees

A tree consists of:
• a set of nodes
• a set of edges, each of which connects a pair of nodes

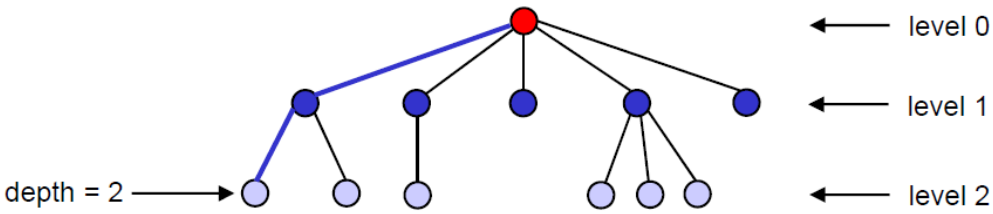
The node at the "top" of the tree is called the root of the tree.



• A node's ancestors are its parent, its parent's parent, etc.
• A node's descendants are its children, their children, etc.
• A leaf node is a node without children.
• An interior node is a node with one or more children.

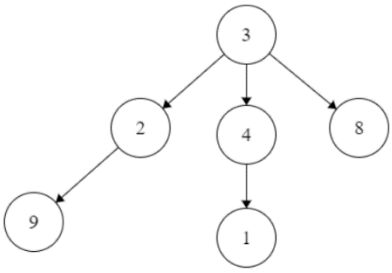


• Nodes with the same depth form a level of the tree.
• The height of a tree is the maximum depth of its nodes.

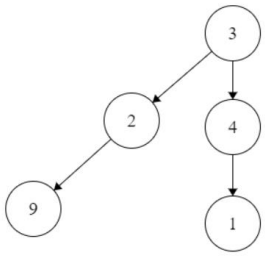


Successor of x = the smallest key which is bigger than x.key (in homework requirement)

Trees

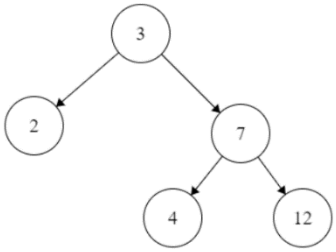


Binary trees



max 2 children

Binary search trees



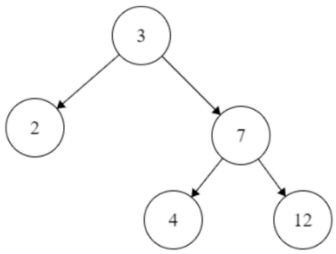
left< root
root<right

Each of the basic operations on a binary search tree runs in O(h) time, where h is the height of the tree

1.Insert into an initially empty binary search tree items with the following keys (in this order): 30, 40, 23, 58, 48, 26, 11, 13.

2. Write the output for inorder, preorder and postorder traversal for the following tree:

Inorder walk: left subtree -> root -> right subtree
Preorder walk: root -> left subtree -> right subtree
Postorder walk: left subtree -> right subtree -> root




```
class Node:
    def __init__(self, data):
        self.left = None
        self.right = None
        self.data = data


def print_tree(node):
    if node:
        print_tree(node.left)
        print(node.data, end=" ")
        print_tree(node.right)
```