

```
public class BST{
    private Node root = null;
    int count;
    private int i = 0;
    BST(){
        root = null;
    }
    public void makeEmpty(){
        root = null;
    }
    public void insert(Comparable x){
        root = insert(x, root);
    }

    public int max(int x, int y){
        return x > y ? x : y;
    }
    private Node insert(Comparable x, Node t){
        if (t == null){
            t = new Node(x, null, null);
            count++;
        }
        else if (x.compareTo(t.data) < 0 )
            t.left = insert(x, t.left);
        else if (x.compareTo(t.data) > 0 )
            t.right = insert(x, t.right);
        else
            ;
        t.height = max(height(t.left) , height(t.right)) + 1;
        return t;
    }
    public void printTree(){
        inPrint(root);
    }
    private void inPrint(Node t){
        if (t != null){
            inPrint(t.left);
            System.out.print (t.data + " ");
            inPrint(t.right);
        }
        if ((i%10) != 0)
            i++;
        else {
            System.out.println();
            i++;
        }
    }
    public int numberOfNodes(){
        return count;
    }
    private int height(Node t) {
        return t == null ? -1 : t.height;
    }
    public int getHeight(){
        return height(root);
    }
}
```