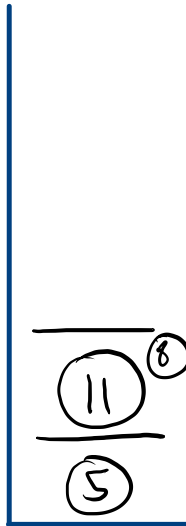




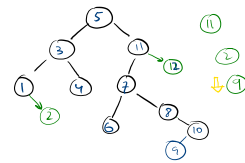
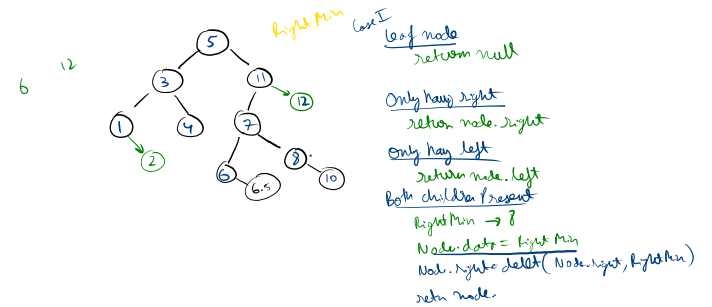
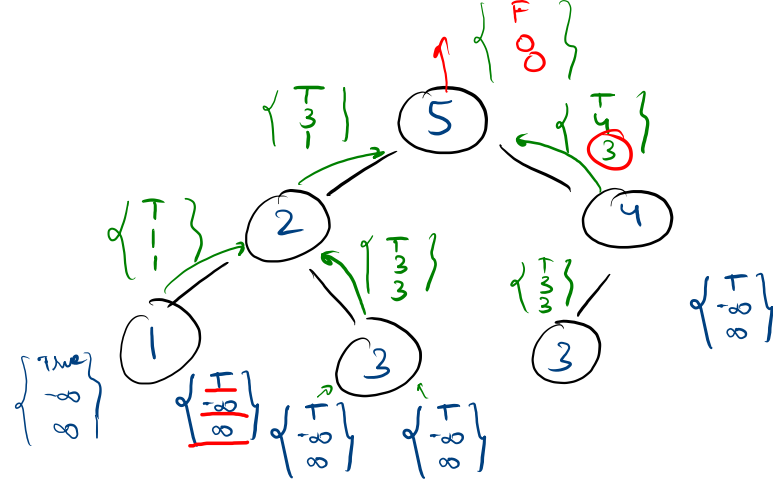
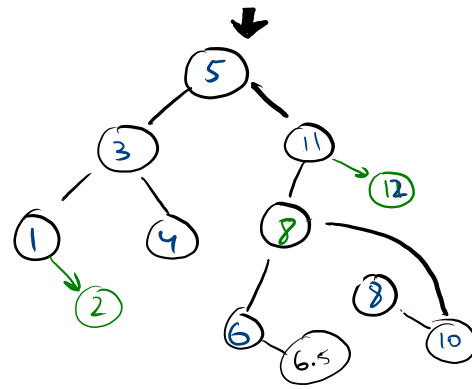
```

public Node deleteNode(Node root, int val){
// WRITE YOUR CODE HERE
if(root == null) return null;
if(root.val == val){
//i am the person who wants to get deleted
//if im a leaf node
if(root.left == null && root.right == null){
return null;
}else if(root.left == null && root.right != null){
//only right child present
return root.right;
}else if(root.left != null && root.right == null){
//only left child
return root.left;
}else{
//both
int rightMin = min(root.right);
root.val = rightMin;
root.right = deleteNode(root.right, rightMin);
}
}else if(val < root.val){
root.left = deleteNode(root.left, val);
}else{
root.right = deleteNode(root.right, val);
}
return root;
}

```

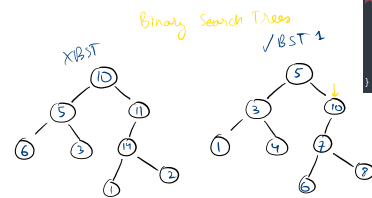
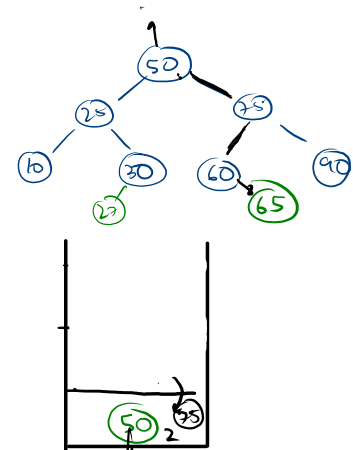


isBST
maxi
mini



Node
func insert(Node, val) <
if (node == null) {
create a node with val = val
return node to parent
}
if (node.val == val) {

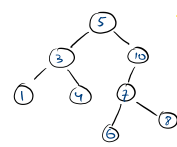
}
if (val < node.data) {
1 -> left new root = insert(node.left, val)
node.left = left new root.
2 -> right new root = insert(node.right, val)
node.right = right new root.
3 -> return node



```

public static int max(Node root) {
// write your code here
// iterative
Node temp = root;
while (temp.right != null) temp = temp.right;
return temp.data;
}

```



I Every parent has
all children smaller
than itself
II In-order traversal
is always
DFS

