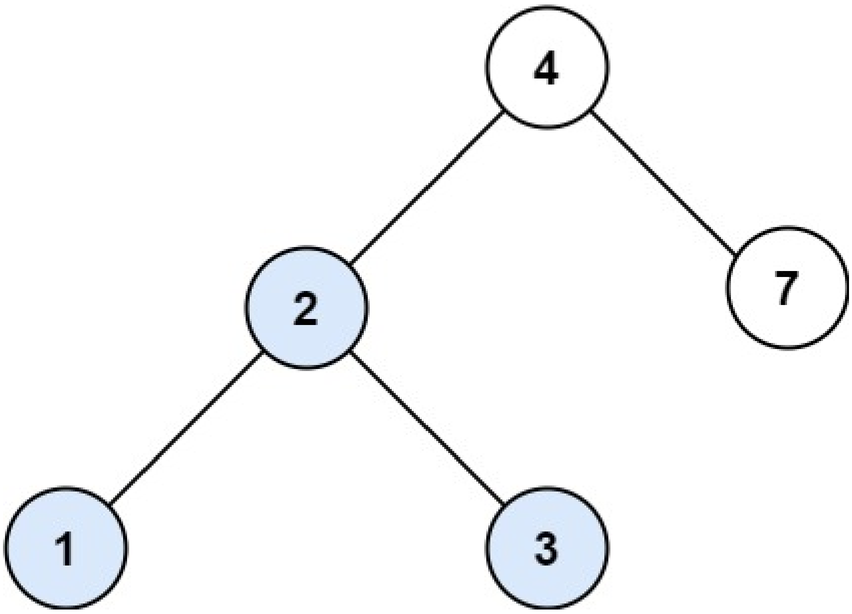
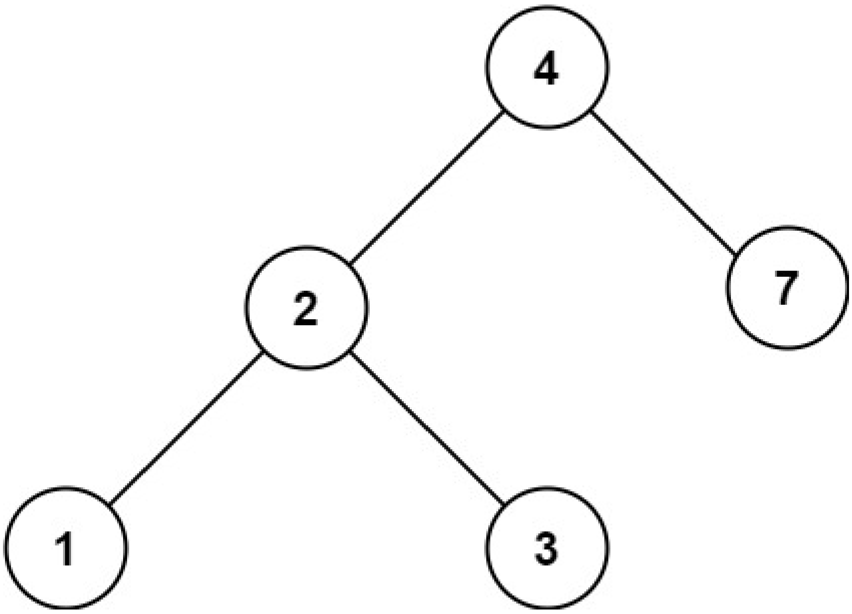
700. Search in a Binary Search Tree

You are given the `root` of a binary search tree (BST) and an integer `val`.

Find the node in the BST that the node's value equals `val` and return the subtree rooted with that node. If such a node does not exist, return `null`.



Input: root = [4,2,7,1,3], val = 2
Output: [2,1,3]



Input: root = [4,2,7,1,3], val = 5
Output: []

```
def searchBST(self, root: TreeNode, val: int) -> TreeNode:
        if root is None:
            return
        if root.val>val:
            lt = self.searchBST(root.left,val)
            return lt
        elif root.val<val:
            rt = self.searchBST(root.right,val)
            return rt
        else:
            return root or lt or rt
```

Insertions a root in bst

```
def insert(root, Key):
    # code here

    if root is None:
        return Node(Key)
    if root.data<Key:
        root.right = insert(root.right, Key)
    elif root.data>Key:
        root.left = insert(root.left, Key)
    return root
```