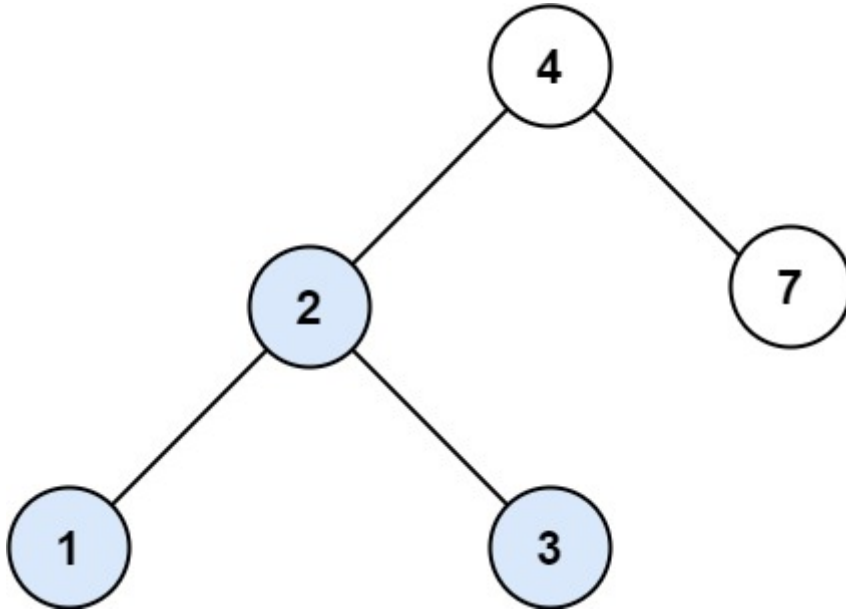


700. Search in a Binary Search Tree

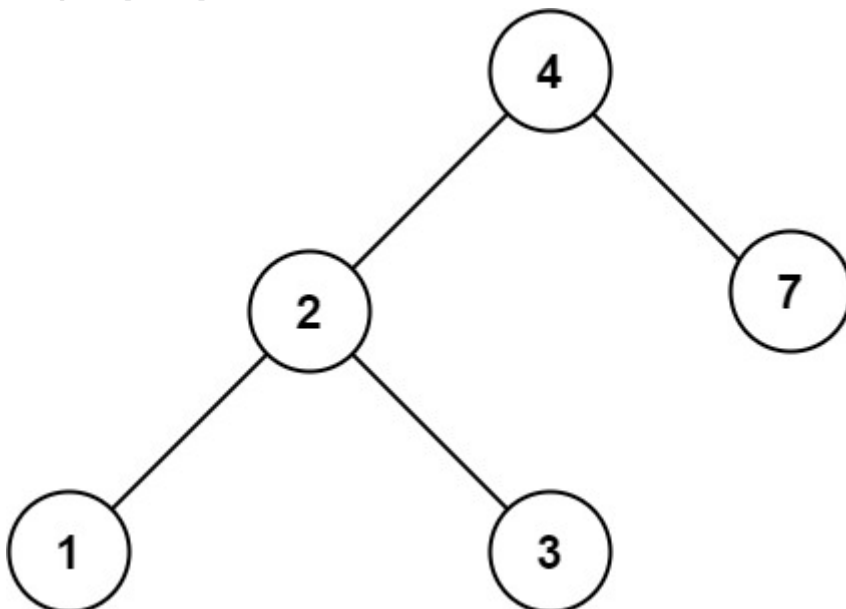
You are given the `root` of a binary search tree (BST) and an integer `val`.

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 None
    node = self.helper(root, val)
    return node
def helper(self, root, val):
    if root is None:
        return
    if root.val > val:
        return self.helper(root.left, val)
    elif root.val < val:
        return self.helper(root.right, val)
    else:
        return root
```