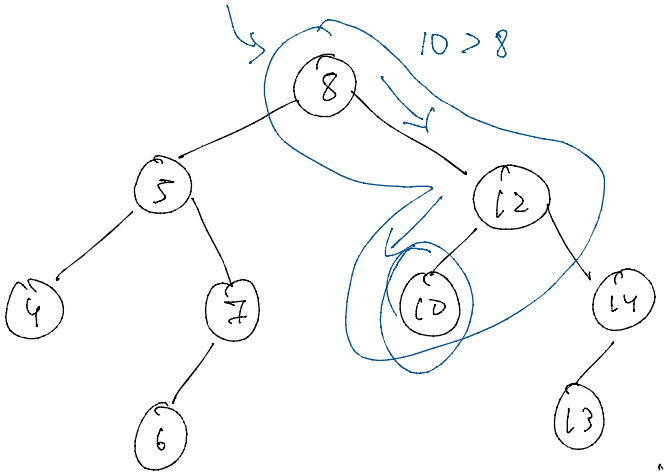
700. Search in a Binary Search Tree

01 April 2022 10:17 AM

Search in BST ⟶ $(\log_2 N)$ height



node = 10

return present

L < N < R

T.C ⇒ $(\log_2 N)$

You won't traverse all the node, either left or right

if node = 9

you go to node = 10

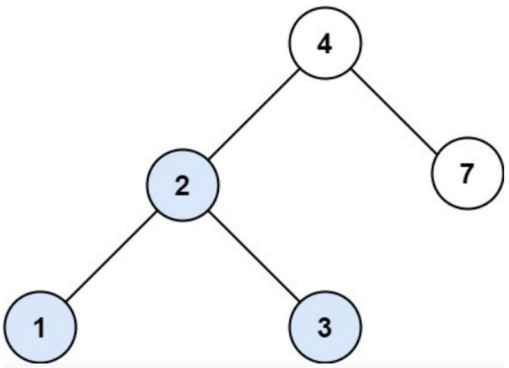
then it should be in left of 10

if it not present then return null.

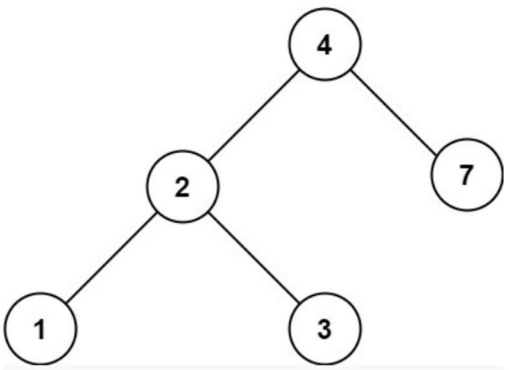
You are given the `root` of a binary search tree (BST) and an integer `val`.

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`.

Example 2:




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




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

```java
class Solution {
    public TreeNode searchBST(TreeNode root, int val) {
        while(root != null && root.val != val){
            // if the root value is less than val then move left else move right
            root = val < root.val ? root.left : root.right;
        }
        return root;
    }
}
```

If it equal it would stop here