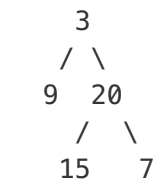
Sum of Left Leaves

Find the sum of all left leaves in a given binary tree.

Example:




```
There are two left leaves in the binary tree, with values 9 and 15 respectively. Re
turn 24.
```

Subscribe to see which companies asked this question

Solution 1

```
public class Solution {
    public int sumOfLeftLeaves(TreeNode root) {
        if(root == null || root.left == null && root.right == null) return 0;

        int res = 0;
        Queue<TreeNode> queue = new LinkedList<>();
        queue.offer(root);

        while(!queue.isEmpty()) {
            TreeNode curr = queue.poll();

            if(curr.left != null && curr.left.left == null && curr.left.right ==
null) res += curr.left.val;
            if(curr.left != null) queue.offer(curr.left);
            if(curr.right != null) queue.offer(curr.right);
        }
        return res;
    }
}
```

written by xietao0221 original link here

Solution 2

```
int sumOfLeftLeaves(TreeNode* root, bool isleft = false) {
    if (!root) return 0;
    if (!root->left && !root->right) return isleft ? root->val : 0;
    return sumOfLeftLeaves(root->left, true) + sumOfLeftLeaves(root->right, false
);
}
```

written by mzchen original link here

Solution 3

base case => node is none
recursive case => Left child is / isn't Leave

```
class Solution(object):
    def sumOfLeftLeaves(self, root):
        if not root: return 0
        if root.left and not root.left.left and not root.left.right:
            return root.left.val + self.sumOfLeftLeaves(root.right)
        return self.sumOfLeftLeaves(root.left) + self.sumOfLeftLeaves(root.right)
# isn't leave
```

EDIT:
Could be 3 Lines, but L2 would be too long.
thanks @tototo's advise!

written by YJL1228 original link here