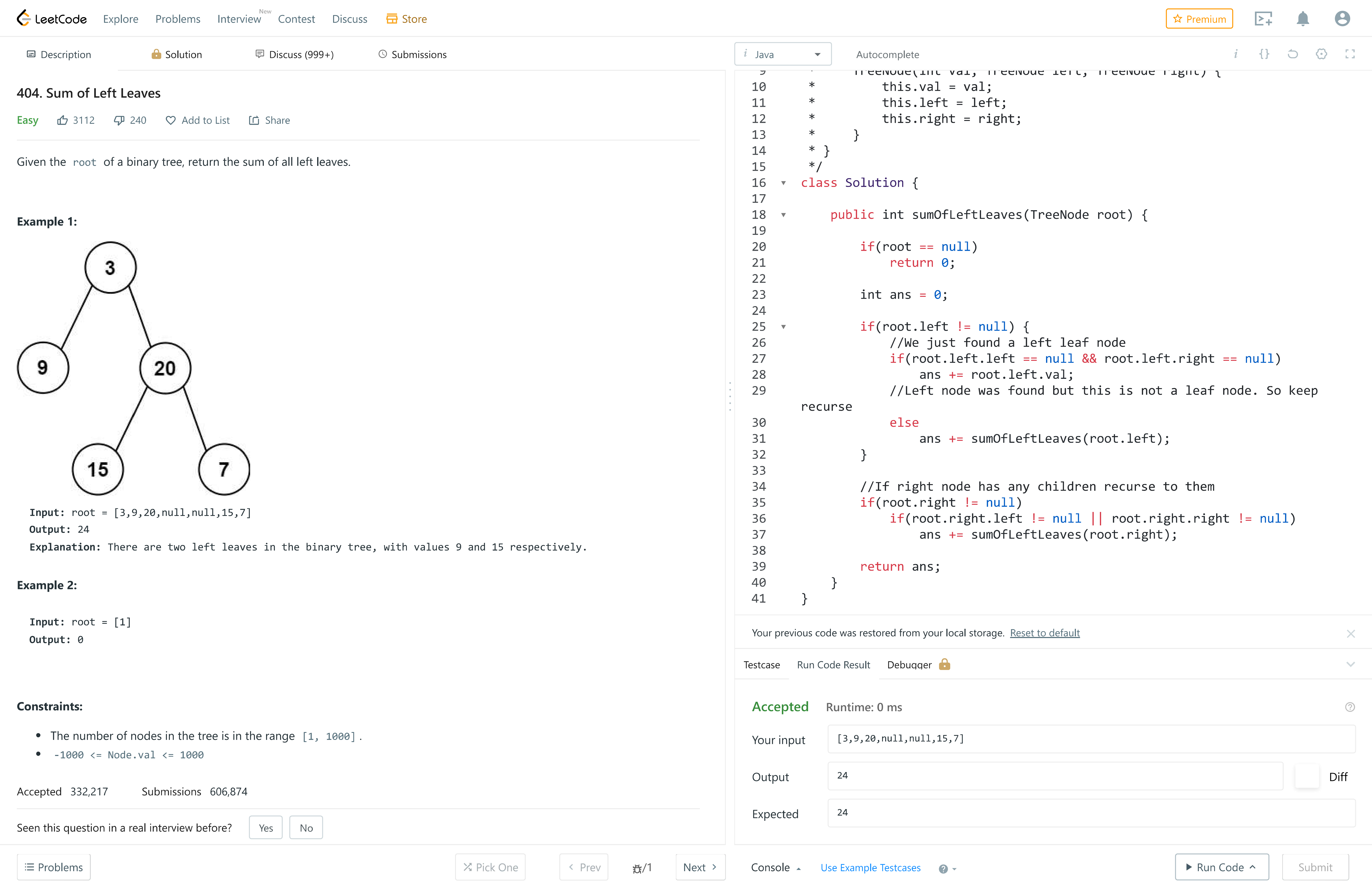

404. Sum of Left Leaves

Given the `root` of a binary tree, return the sum of all left leaves.

Example 1:

```
Input: root = [3,9,20,null,null,15,7]
Output: 24
Explanation: There are two left leaves in the binary tree, with values 9 and 15 respectively.
```

Example 2:

```
Input: root = [1]
Output: 0
```

Constraints:

- The number of nodes in the tree is in the range `[1, 1000]`.
- `-1000 <= Node.val <= 1000`

Accepted 332,217   Submissions 606,874

Seen this question in a real interview before? Yes No

```java
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */
class Solution {

    public int sumOfLeftLeaves(TreeNode root) {

        if(root == null)
            return 0;

        int ans = 0;

        if(root.left != null) {
            //We just found a left leaf node
            if(root.left.left == null && root.left.right == null)
                ans += root.left.val;
            //Left node was found but this is not a leaf node. So keep recurse
            else
                ans += sumOfLeftLeaves(root.left);
        }

        //If right node has any children recurse to them
        if(root.right != null)
            if(root.right.left != null || root.right.right != null)
                ans += sumOfLeftLeaves(root.right);

        return ans;
    }
}
```

Your previous code was restored from your local storage. Reset to default

Accepted   Runtime: 0 ms

Your input `[3,9,20,null,null,15,7]`

Output `24`

Expected `24`