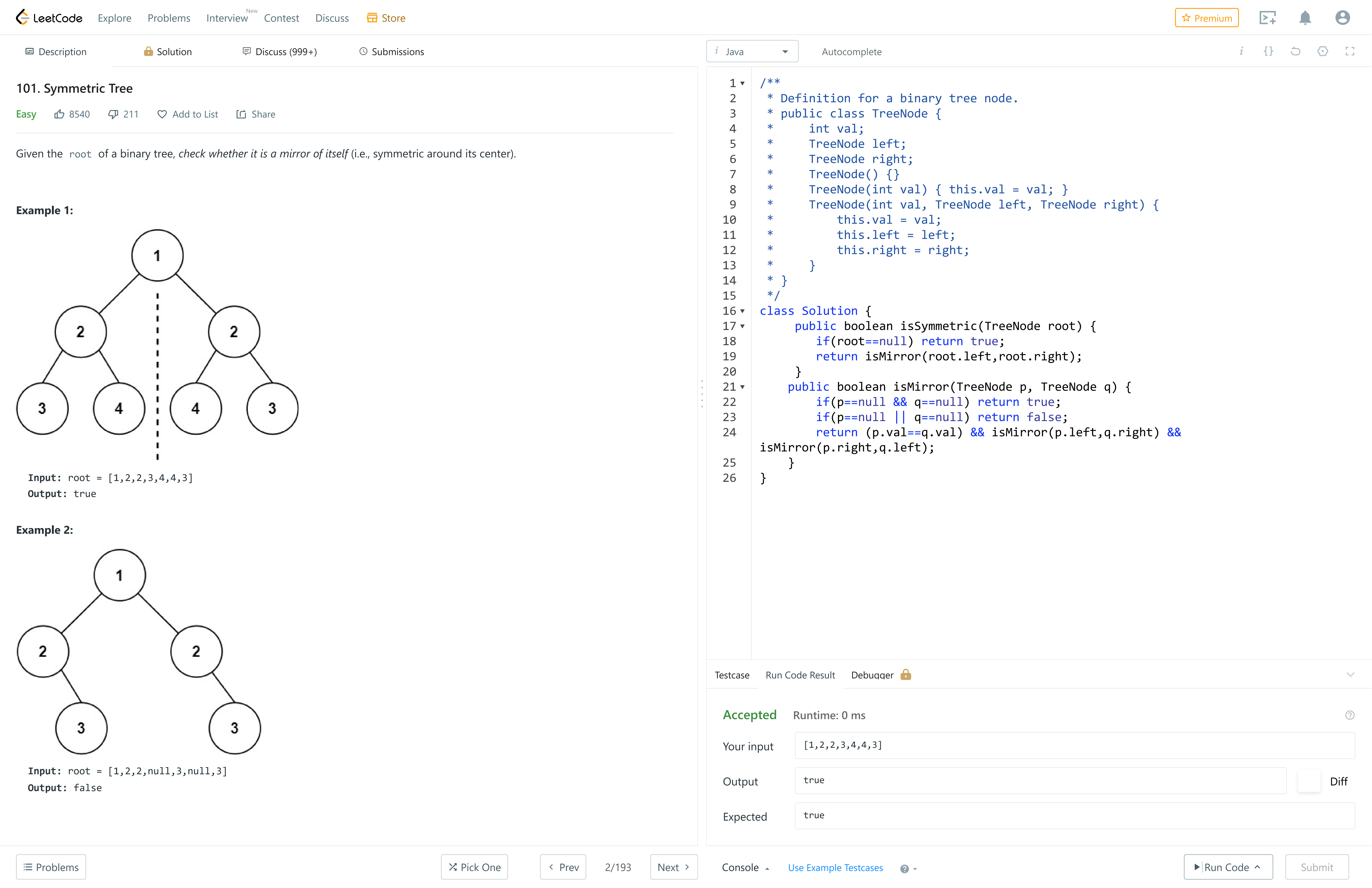

101. Symmetric Tree

Given the `root` of a binary tree, *check whether it is a mirror of itself* (i.e., symmetric around its center).

Example 1:

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

Example 2:

```
Input: root = [1,2,2,null,3,null,3]
Output: false
```

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */
class Solution {
     public boolean isSymmetric(TreeNode root) {
        if(root==null) return true;
        return isMirror(root.left,root.right);
     }
    public boolean isMirror(TreeNode p, TreeNode q) {
        if(p==null && q==null) return true;
        if(p==null || q==null) return false;
        return (p.val==q.val) && isMirror(p.left,q.right) && isMirror(p.right,q.left);
    }
}
```

Accepted Runtime: 0 ms

Your input [1,2,2,3,4,4,3]

Output true

Expected true