

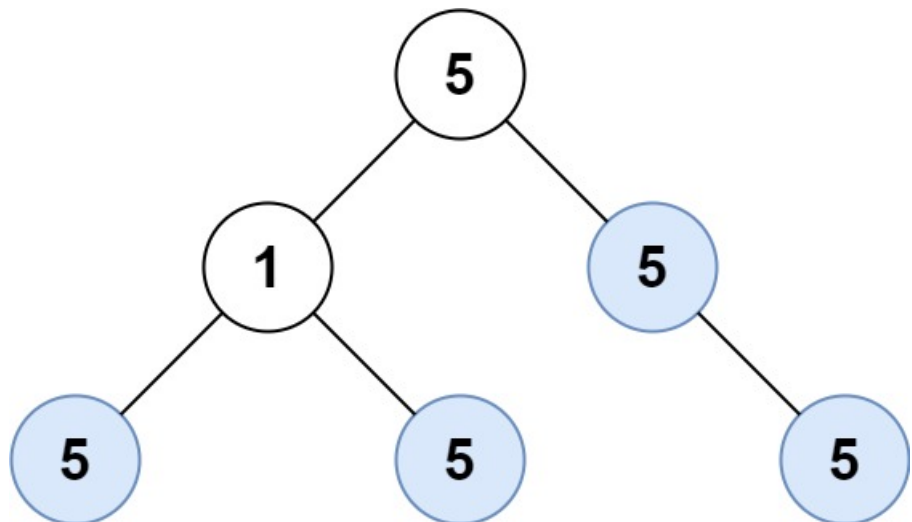
250. Count Unival Subtrees

Medium👤 823🗨️ 246💖 Add to List🔖 Share

Given the `root`, of a binary tree, return the number of **uni-value** subtrees.

A **uni-value subtree** means all nodes of the subtree have the same value.

Example 1:



Input: root = [5,1,5,5,5,null,5]
Output: 4

Example 2:

Input: root = []
Output: 0

Example 3:

Input: root = [5,5,5,5,5,null,5]
Output: 6

Constraints:

- The numbrt of the node in the tree will be in the range `[0, 1000]`.
- `-1000 <= Node.val <= 1000`

Accepted 103,064 | Submissions 189,938

Seen this question in a real interview before?

Companies👤 1

0 ~ 6 months6 months ~ 1 year1 year ~ 2 years

Google 2

Related Topics

TreeDepth-First SearchBinary Tree

Similar Questions

Subtree of Another TreeEasy

Longest Univalue PathMedium

```
1 public class Solution {
2     int count = 0;
3     boolean is_valid_part(TreeNode node, int val) {
4
5         // considered a valid subtree
6         if (node == null) return true;
7
8         // check if node.left and node.right are univalue subtrees of value node.val
9         // note that ! short circuits but ! does not - both sides of the or get evaluated with ! so we explore all possible routes
10        if (!is_valid_part(node.left, node.val) || !is_valid_part(node.right, node.val)) return false;
11
12        // if it passed the last step then this a valid subtree - increment
13        count++;
14
15        // at this point we know that this node is a univalue subtree of value node.val
16        // pass a boolean indicating if this is a valid subtree for the parent node
17        return node.val == val;
18    }
19    public int countUnivalSubtrees(TreeNode root) {
20        is_valid_part(root, 0);
21        return count;
22    }
23 }
```