Problem 250: Count Univalue Subtrees

Problem Information

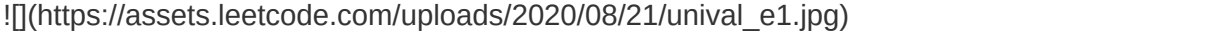
Difficulty: Medium
Acceptance Rate: 57.36%
Paid Only: Yes
Tags: Tree, Depth-First Search, Binary Tree

Problem Description

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

Code Snippets

C++:

```
/**
* Definition for a binary tree node.
* struct TreeNode {
* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode() : val(0), left(nullptr), right(nullptr) {}
* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
* TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
* };
*/
class Solution {
public:
int countUnivalSubtrees(TreeNode* root) {

}
};
```

Java:

```
/**
* 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 int countUnivalSubtrees(TreeNode root) {

}
}
```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, val=0, left=None, right=None):
# self.val = val
# self.left = left
# self.right = right
class Solution:
def countUnivalSubtrees(self, root: Optional[TreeNode]) -> int:
```