Problem 637: Average of Levels in Binary Tree

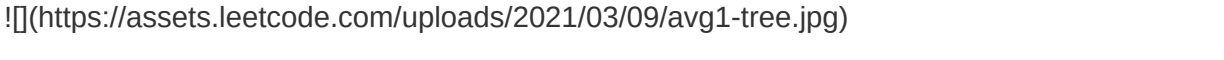
Problem Information

Difficulty: Easy
Acceptance Rate: 74.54%
Paid Only: No
Tags: Tree, Depth-First Search, Breadth-First Search, Binary Tree

Problem Description

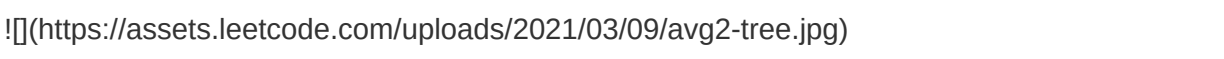
Given the `root` of a binary tree, return _the average value of the nodes on each level in the form of an array_. Answers within `10-5` of the actual answer will be accepted.

Example 1:



Input: root = [3,9,20,null,null,15,7] **Output:** [3.00000,14.50000,11.00000] Explanation: The average value of nodes on level 0 is 3, on level 1 is 14.5, and on level 2 is 11. Hence return [3, 14.5, 11].

Example 2:



Input: root = [3,9,20,15,7] **Output:** [3.00000,14.50000,11.00000]

Constraints:

* The number of nodes in the tree is in the range `[1, 104]`. * `-231 <= Node.val <= 231 - 1`

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:
vector<double> averageOfLevels(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 List<Double> averageOfLevels(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 averageOfLevels(self, root: Optional[TreeNode]) -> List[float]:
```