Problem 1161: Maximum Level Sum of a Binary Tree

Problem Information

Difficulty: Medium
Acceptance Rate: 67.47%
Paid Only: No
Tags: Tree, Depth-First Search, Breadth-First Search, Binary Tree

Problem Description

Given the `root` of a binary tree, the level of its root is `1`, the level of its children is `2`, and so on.

Return the **smallest** level `x` such that the sum of all the values of nodes at level `x` is **maximal**.

Example 1:

Input: root = [1,7,0,7,-8,null,null] **Output:** 2 **Explanation:** Level 1 sum = 1. Level 2 sum = 7 + 0 = 7. Level 3 sum = 7 + -8 = -1. So we return the level with the maximum sum which is level 2.

Example 2:

Input: root = [989,null,10250,98693,-89388,null,null,null,-32127] **Output:** 2

Constraints:

* The number of nodes in the tree is in the range `[1, 104]`. * `-105 <= Node.val <= 105`

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 maxLevelSum(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 maxLevelSum(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 maxLevelSum(self, root: Optional[TreeNode]) -> int:
```