Problem 3157: Find the Level of Tree with Minimum Sum

Problem Information

Difficulty: Medium
Acceptance Rate: 69.05%
Paid Only: Yes
Tags: Tree, Depth-First Search, Breadth-First Search, Binary Tree

Problem Description

Given the root of a binary tree `root` where each node has a value, return the level of the tree that has the **minimum** sum of values among all the levels (in case of a tie, return the **lowest** level).

Note that the root of the tree is at level 1 and the level of any other node is its distance from the root + 1.

Example 1:

Input: root = [50,6,2,30,80,7]

Output: 2

Explanation:

Example 2:

Input: root = [36,17,10,null,null,24]

Output: 3

Explanation:

Example 3:

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

Output: 1

Explanation:

Constraints:

* The number of nodes in the tree is in the range `[1, 105]`. * `1 <= Node.val <= 109`

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 minimumLevel(TreeNode* root) {

 }
};
```

Java:

```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 minimumLevel(TreeNode root) {


}
}
```

Python3:

```python
# 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 minimumLevel(self, root: Optional[TreeNode]) -> int:
```