Problem 671: Second Minimum Node In a Binary Tree

Problem Information

Difficulty: Easy
Acceptance Rate: 45.75%
Paid Only: No
Tags: Tree, Depth-First Search, Binary Tree

Problem Description

Given a non-empty special binary tree consisting of nodes with the non- negative value, where each node in this tree has exactly `two` or `zero` sub- node. If the node has two sub-nodes, then this node's value is the smaller value among its two sub-nodes. More formally, the property `root.val = min(root.left.val, root.right.val)` always holds.

Given such a binary tree, you need to output the **second minimum** value in the set made of all the nodes' value in the whole tree.

If no such second minimum value exists, output -1 instead.

Example 1:

Input: root = [2,2,5,null,null,5,7] **Output:** 5 **Explanation:** The smallest value is 2, the second smallest value is 5.

Example 2:

Input: root = [2,2,2] **Output:** -1 **Explanation:** The smallest value is 2, but there isn't any second smallest value.

Constraints:

* The number of nodes in the tree is in the range `[1, 25]`. * `1 <= Node.val <= 231 - 1` * `root.val == min(root.left.val, root.right.val)` for each internal node of the tree.

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