Problem 988: Smallest String Starting From Leaf

Problem Information

Difficulty: Medium
Acceptance Rate: 61.02%
Paid Only: No
Tags: String, Backtracking, Tree, Depth-First Search, Binary Tree

Problem Description

You are given the `root` of a binary tree where each node has a value in the range `[0, 25]` representing the letters `'a'` to `'z'`.

Return _the**lexicographically smallest** string that starts at a leaf of this tree and ends at the root_.

As a reminder, any shorter prefix of a string is **lexicographically smaller**.

* For example, `"ab"` is lexicographically smaller than `"aba"`.

A leaf of a node is a node that has no children.

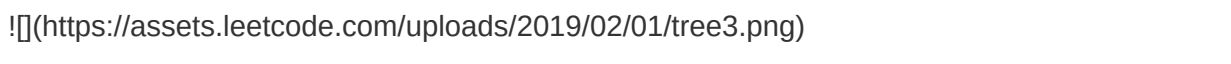
Example 1:

Input: root = [0,1,2,3,4,3,4] **Output:** "dba"

Example 2:

Input: root = [25,1,3,1,3,0,2] **Output:** "adz"

Example 3:



Input: root = [2,2,1,null,1,0,null,0] **Output:** "abc"

Constraints:

* The number of nodes in the tree is in the range `[1, 8500]`. * `0 <= Node.val <= 25`

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:
string smallestFromLeaf(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 String smallestFromLeaf(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 smallestFromLeaf(self, root: Optional[TreeNode]) -> str:
```