Root To Leaf Paths Sum

solved by	Senan
Platform	GeeksForGeeks
difficulty	Medium
tags	DFS Tree
language	C++
solved on	@06/11/2024
link	https://www.geeksforgeeks.org/problems/root-to-leaf-paths-sum/1
Completion	✓

Intuition

The problem is about calculating the sum of all numbers formed by root-to-leaf paths in a binary tree. Each path from the root to a leaf represents a number formed by concatenating the node values along the path. The goal is to compute the sum of all such numbers.

Approach

1. Use Depth-First Search (DFS) to traverse the binary tree recursively.
2. Pass the current `sum` formed by concatenating node values to each recursive call.
3. When a leaf node (a node with no children) is reached, add the current path sum to the total sum.
4. Continue traversing the left and right subtrees, updating the path sum accordingly.
5. Return the accumulated sum after all paths have been processed.

Complexity

Time Complexity:

- **O(n)**, where `n` is the number of nodes in the tree. Each node is visited exactly once in the DFS traversal.

Space Complexity:

- **O(h)**, where `h` is the height of the tree. This represents the space taken by the call stack during recursion. In the worst case (a skewed tree), this could be **O(n)**.

Code

```
class Solution {
    void dfs(Node* root, int sum, int &pathSum) {
        if (root == NULL) return;
        int newSum = 10 * sum + root->data;
        if (root->left == NULL && root->right == NULL) {
            pathSum += newSum;
            return;
        }
        dfs(root->left, newSum, pathSum);
        dfs(root->right, newSum, pathSum);
    }

  public:
    int treePathsSum(Node *root) {
        int totalSum = 0;
        dfs(root, 0, totalSum);
        return totalSum;
    }
};
```