Problem 1325: Delete Leaves With a Given Value

Problem Information

Difficulty: Medium
Acceptance Rate: 77.24%
Paid Only: No
Tags: Tree, Depth-First Search, Binary Tree

Problem Description

Given a binary tree `root` and an integer `target`, delete all the **leaf nodes** with value `target`.

Note that once you delete a leaf node with value `target`**,** if its parent node becomes a leaf node and has the value `target`, it should also be deleted (you need to continue doing that until you cannot).

Example 1:

![](https://assets.leetcode.com/uploads/2020/01/09/sample_1_1684.png)

Input: root = [1,2,3,2,null,2,4], target = 2 **Output:** [1,null,3,null,4] **Explanation:** Leaf nodes in green with value (target = 2) are removed (Picture in left). After removing, new nodes become leaf nodes with value (target = 2) (Picture in center).

Example 2:

![](https://assets.leetcode.com/uploads/2020/01/09/sample_2_1684.png)

Input: root = [1,3,3,3,2], target = 3 **Output:** [1,3,null,null,2]

Example 3:

![](https://assets.leetcode.com/uploads/2020/01/15/sample_3_1684.png)

Input: root = [1,2,null,2,null,2], target = 2 **Output:** [1] **Explanation:** Leaf nodes in green with value (target = 2) are removed at each step.

Constraints:

* The number of nodes in the tree is in the range `[1, 3000]`. * `1 <= Node.val, target <= 1000`

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:
TreeNode* removeLeafNodes(TreeNode* root, int target) {

}
};
```

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 TreeNode removeLeafNodes(TreeNode root, int target) {

}
}
```

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