156. Binary Tree Upside Down

Question My Submissions (/problems/binary-tree-upside-down/submissions/)

Total Accepted: **7960** Total Submissions: **21522** Difficulty: **Medium**

> Given a binary tree where all the right nodes are either leaf nodes with a sibling (a left node that shares the same parent node) or empty, flip it upside down and turn it into a tree where the original right nodes turned into left leaf nodes. Return the new root.
>
> For example:
>
> Given a binary tree `{1,2,3,4,5}`,
>
> ```
>     1
>    / \
>   2   3
>  / \
> 4   5
> ```
>
> return the root of the binary tree `[4,5,2,#,#,3,1]`.
>
> ```
>    4
>   / \
>  5   2
>     / \
>    3   1
> ```
>
> confused what `"{1,#,2,3}"` means? > read more on how binary tree is serialized on OJ.
>
> Show Tags
>
> Show Similar Problems

Have you met this question in a real interview? Yes No

Discuss (/discuss/questions/oj/binary-tree-upside-down)

C++

✉ Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

```
/**
 * Definition for a binary tree node.
```

```
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    TreeNode* upsideDownBinaryTree(TreeNode* root) {
        
    }
};
```

Custom Testcase □

Run Code

Submit Solution

Frequently Asked Questions (/faq/) | Terms of Service (/tos/)

Privacv