156. Binary Tree Upside Down

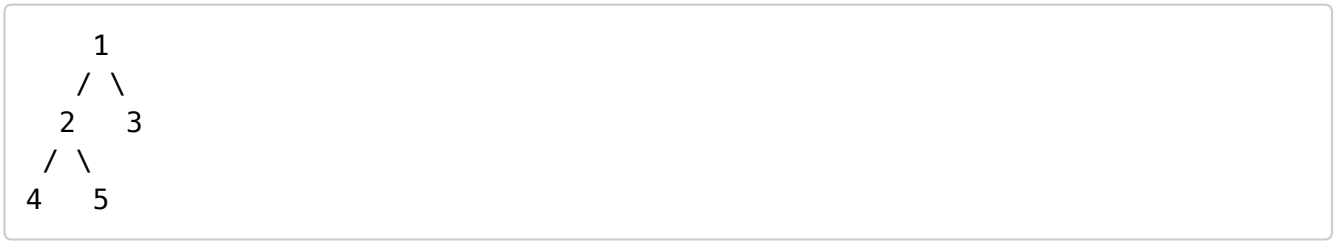
Question | My Submissions (/problems/binary-tree-upside-down/submissions/)

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

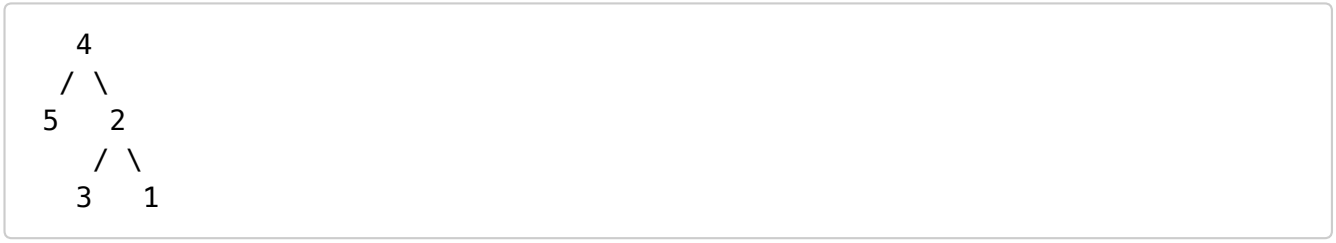
Given a binary tree where all the right nodes are either leaf nodes with a sibling (a left node that shares the same parent node) or empty, flip it upside down and turn it into a tree where the original right nodes turned into left leaf nodes. Return the new root.

For example:

Given a binary tree `{1,2,3,4,5}`,



return the root of the binary tree `[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++

```
1  /** ✉ Send Feedback (mailto:admin@leetcode.com?subject=Feedback)
2   * Definition for a binary tree node.
```

```
 3   * struct TreeNode {
 4   *     int val;
 5   *     TreeNode *left;
 6   *     TreeNode *right;
 7   *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 8   * };
 9   */
10  class Solution {
11  public:
12      TreeNode* upsideDownBinaryTree(TreeNode* root) {
13
14      }
15  };
```

Custom Testcase

Run Code | Submit Solution