117 375

156. Binary Tree Upside Down

Notes

Description (/problems/binary-tree-upside-down/description/) Hints (/problems/binary-tree-upside-down/hints/)

Java recursive (O(logn) space) and iterative solutions (O(1) space) with ex

14.7K
VIEWS

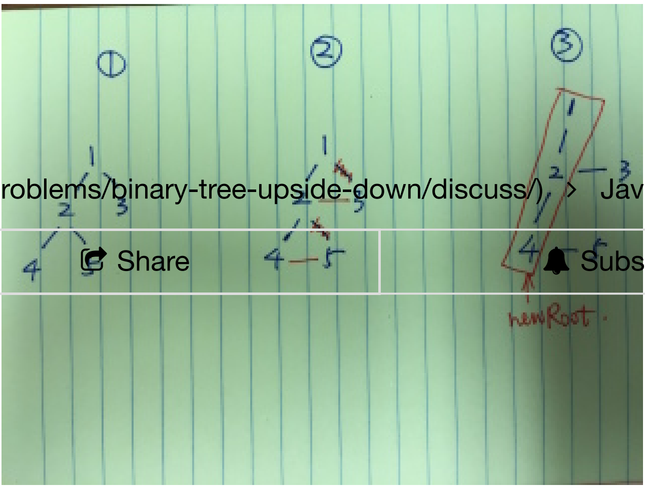
214

Last Edit: September 22, 2018 11:40 AM

(/yfcheng) yfcheng (/yfcheng) ★
2269

This is not a very intuitive problem for me, I have to spent quite a while drawing figures to understand it. As shown in the figure, 1 shows the original tree, you can think about it as a comb, with 1, 2, 4 form the bone, and 3, 5 as the teeth. All we need to do is flip the teeth direction as shown in figure 2. We will remove the link 1--3, 2--5, and add link 2--3, and 4--5. And node 4 will be the new root.

As the recursive solution, we will keep recurse on the left child and once we are are done, we found the newRoot, which is 4 for this case. At this point, we will need to set the new children for node 2, basically the new left node is 3, and right node is 1. Here is the recursive solution:



☰ (/problems/binary-tree-upside-down/discuss/) › Java recursive (O(logn) space) and iterative solutions (O(1)…

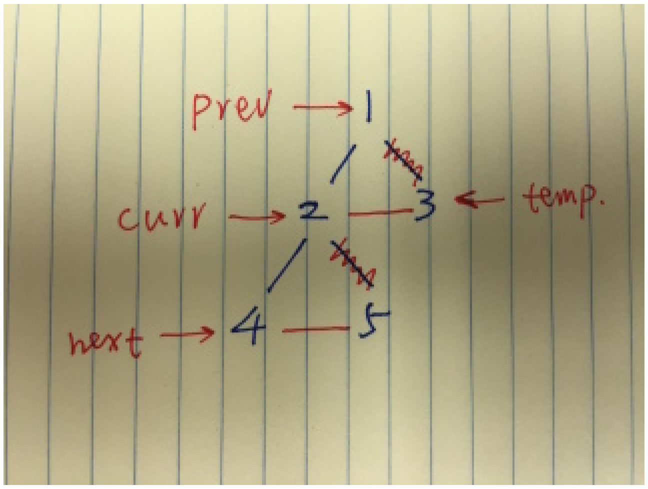
Share | Subscribe | Report

Recursive:

```
public TreeNode upsideDownBinaryTree(TreeNode root) {
    if(root == null || root.left == null) {
        return root;
    }

    TreeNode newRoot = upsideDownBinaryTree(root.left);
    root.left.left = root.right;   // node 2 left children
    root.left.right = root;         // node 2 right children
    root.left = null;
    root.right = null;
    return newRoot;
}
```

For the iterative solution, it follows the same thought, the only thing we need to pay attention to is to save the node information that will be overwritten.



```
public TreeNode upsideDownBinaryTree(TreeNode root) {
    TreeNode curr = root;
    TreeNode next = null;
    TreeNode temp = null;
    TreeNode prev = null;

    while(curr != null) {
        next = curr.left;

        // swapping nodes now, need temp to keep the previous right child
        curr.left = temp;
        temp = curr.right;
        curr.right = prev;

        prev = curr;
        curr = next;
    }
    return prev;
}
```

Comments: 11

Sort By ▾

Type comment here... (Markdown is supported)

👁 **Preview** | Post

Yunwei_Qiu (/yunwei_qiu) ★ 45 ◷ July 20, 2018 6:17 PM

Thanks for your posting. I think this problem is very similar to the problem 206. Reverse Linked List. I recommend u guys can practice these Recursive and iterative technology on both questions. Then u will find the rule :)
Let me show u the magic:
The Recursive on both problems:

Read More

10 ^ v | Share | Reply

SHOW 1 REPLY

anon2453 (/anon2453) ★ 1 ◷ February 28, 2018 9:59 PM

class Solution {
public TreeNode upsideDownBinaryTree(TreeNode root) {
if (root == null) {
return root;
}

Read More

1 ^ v | Share | Reply

GritShiva (/gritshiva) ★ 4 ◷ January 3, 2018 10:19 AM

Thanks for sharing, the picture illustration helped a lot.

0 ^ v | Share | Reply

lbrobinho (/lbrobinho) ★ 12 ◷ November 10, 2017 9:56 AM

Iterative way is from top to bottom, recursive way is from bottom to top.

2 ^ v | Share | Reply

vegito2002@gmail.com (/vegito2002gmailcom) ★ 534 ◷ August 30, 2017 2:56 PM

Sharing my very similar iterative solution here:

Read More

3 ˄ ˅ | Share | Reply

new2500 (/new2500) ★ 151 🕘 May 27, 2017 1:21 PM ⋮

Thumb UP. Thank you for the clear explanation and process of thoughts.

I think for the recursive space, Should it be O(depth of tree) than O(logN) since it's random binary tree?

Notes

0 ˄ ˅ | Share | Reply

sonali3 (/sonali3) ★ 8 🕘 May 26, 2017 10:39 AM ⋮

Thanks for explaining using diagram. This helps a lot.

0 ˄ ˅ | Share | Reply

knowledge2thepeople (/knowledge2thepeople) ★ 128 🕘 May 17, 2017 9:55 PM ⋮

@yfcheng Nice iterative solution. Here's my iterative solution. I'm using more variables than I need:

Read More

0 ˄ ˅ | Share | Reply

GardenAAA (/gardenaaa) ★ 86 🕘 February 3, 2017 1:32 PM ⋮

Really smart answer and detailed explanation. Thanks for sharing.

2 ˄ ˅ | Share | Reply

feyhi (/feyhi) ★ 30 🕘 December 22, 2016 3:28 AM ⋮

Nice write-up, but since the tree is not a typical tree (more like a trunk), I don't think the recursive solution is `O(logn)` space -- it should be `O(n)`.

12 ˄ ˅ | Share | Reply

SHOW 1 REPLY

‹ 1 2 ›

Contact Us (/support/) | Students (/students) | Frequently Asked Questions (/faq/) | Terms of Service (/terms/) |
Privacy Policy (/privacy/)

United States (/region/)

Notes