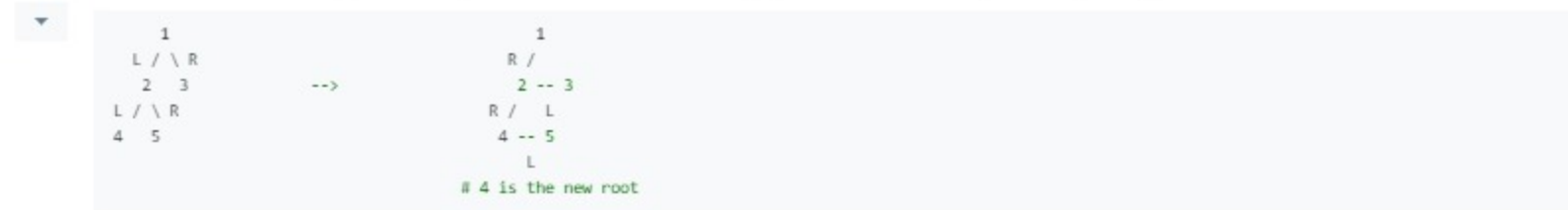


[Back](#) Concise python iterative solution with O(1) space

Balez 4 Last Edit: March 25, 2020 11:12 PM 45 VIEWS

3 The transformation from the input to the output is easier to understand in this way (L and R denote left and right child respectively):



In this viewpoint, the problem can be solved by traversing the leftmost root-to-leaf path once (which is 1 -> 2 -> 4 in the above example) meanwhile modifying the parent-child links. The modification in each iteration is like:



The python code:

```

class Solution:
    def upsideDownBinaryTree(self, root: TreeNode) -> TreeNode:
        # start from root
        cur, pre, preRight = root, None, None
        while cur:
            # temporarily store 'cur.left' and 'cur.right' ('cur.left' will be next 'cur')
            temp1, temp2 = cur.left, cur.right
            # modify parent-child links
            cur.left, cur.right = preRight, pre
            # go to next iteration
            cur, pre, preRight = temp1, cur, temp2
        return pre
  
```

Complexity: O(N) time and O(1) space, where N is the number of nodes in the input tree.

python iterative constant space

Comments: 2

Best Most Votes Newest to Oldest Oldest to Newest

Type comment here... (Markdown is supported)

Post

marapareddy 0 April 23, 2020 6:53 AM

Simply superb. The comments and explanation are really good.

0 Reply

hwc0919 24 April 3, 2020 11:25 AM

Fantastic

0 Reply