喜刷刷

Monday, November 17, 2014

[LeetCode新题] Binary Tree Upside Down

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
```

思路：

LeetCode最近出了收费模式的新题，需要买他们出的电子书才能做。不过既然已经免费使用了它家这么多资源，就当买本书做点贡献。

这题第一眼看上去觉得没头绪，不知道怎么上下翻转和左右翻转。但在纸上画几个例子就清楚了。以题目的例子来解释：

1. 对于一个parent来说，加入有right node，必须得有left node。而有left node，right node可以为空。而right node必须为叶子节点。所以该树每层至多有2个节点，并且2节点有共同的parent。

2. 所以对于最底层来说，必有一个left node，而这个left node则为整个新树的根 —— 例子中的4为最底层的左节点，最后成为新树的root。

3. 原树的根节点，变为了新树的最右节点。

3. 对于子树1 2 3来说，需要在以2为根的子树2 4 5建立成新树4 5 2后，插入到新树的最右节点2下面。原树的根节点root为left child，原树root->right为新树的left nnode

递归实现：

```
1 class Solution {
2 public:
3     TreeNode *upsideDownBinaryTree(TreeNode *root) {
4         TreeNode *temp, *newRoot = NULL;
5         temp = buildUpsideDownBT(root, newRoot);
6         return newRoot;
7     }
8
9     TreeNode *buildUpsideDownBT(TreeNode *root, TreeNode *&newRoot) {
10        if(!root) return root;
11        if(!root->left && !root->right) {
12            newRoot = root;
13            return root;
14        }
15        TreeNode *parent = buildUpsideDownBT(root->left, newRoot);
16        parent->left = root->right;
17        parent->right = root;
18        root->left = root->right = NULL;
19        return parent->right;
20    }
21 };
```

总结：

1. 这个递归的核心是，每次建立好一个新的子树后，要返回新子树的最右节点（**In 19)**，以便上层的节点可以接回到这个节点的下面。

2. 但如果只返回最右节点，则我们无法知道最后整个新树的根在哪里。所以再**base case**里必须给新根赋值**(In 12)**

3. 每次需要**reset**最右节点的**left/right node**，否则最后一层递归，递归到例子中的**1**节点时，返回前**1**节点的**left/right node**仍然为原来的值，而并不为**NULL**。

Posted by Yanbing Shi at 8:57 AM +2 Recommend this on Google

Labels: algorithm, binary tree, data structure, Leetcode, recursive

5 comments:

Hejia Pan November 22, 2014 at 9:00 PM

觉得这题就是后序变成层序.

Reply

Replies

Ann GUO September 16, 2015 at 7:05 PM

赞！

Ann GUO September 16, 2015 at 7:10 PM

赞！

Charles Hu November 5, 2015 at 10:46 AM

我也写了一个. 每次只返回根. 但是在递归前记录要链接的节点. 在自己电脑上运行了一下,没发现什么问题. 大侠们确认一下?

```
TreeNode* upDown(TreeNode*root){
if (!root) return nullptr;
if (!root->left) return root;

assert(root->left);
TreeNode* l= root->left;
TreeNode* r= root->right;
TreeNode* newH= upDown(root->left);
l->right= root;
l->left= upDown(r);
root->left=0;
root->right=0;
if (r) { r->left=0; r->right=0;}
return newH;
}
```

Reply

gary zhang July 18, 2016 at 11:45 PM

不是很理解题目。。这样子满足题目的条件吗？

1
/ \
3 2
/ \
5 4

Reply

Enter your comment...

Comment as: Select profile...

Publish Preview

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)