114 Flatten Binary Tree to Linked List

Given a binary tree, flatten it to a linked list in-place.

For example, Given

```
        1
       / \
      2   5
     / \   \
    3   4   6
```

```
   1
    \
     2
      \
       3
        \
         4
          \
           5
            \
             6
```

[Hints:

If you notice carefully in the flattened tree, each node's right child points to the next node of a pre-order traversal.](https://leetcode.com/problems/flatten-binary-tree-to-linked-list/description/#)

用一个栈保存数据即可。但是需要用到O(n) 的空间。

```
/**
 * 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:

    void preorder(TreeNode* root, stack<TreeNode* >& stack){
        if( root == NULL )  return;
        stack.push(root);
```

```
        preorder(root -> left, stack);
        preorder(root -> right, stack);
    }

    void flatten(TreeNode* root) {
        if(root == NULL) return;

        stack<TreeNode *>  stack;       /// 用来保存中序遍历的节点
        preorder(root, stack);
        TreeNode* pre = stack.top(); stack.pop();

        while( !stack.empty() ){
            root = stack.top(); stack.pop();
            root -> left = NULL;
            root -> right = pre;
            pre = root;
        }


    }
};
```