114. Flatten Binary Tree to Linked List

题目描述：https://leetcode.com/problems/flatten-binary-tree-to-linked-list/

> 把二叉树转成链表。
> 例如：

```
Given:

     1
    / \
   2   5
  / \   \
 3   4   6
The flattened tree should look like:
1
 \
  2
   \
    3
     \
      4
       \
        5
         \
          6
```

解题思路：

right = preOrder

代码 递归：

```
/**
 * 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 flat(TreeNode* root) {
        if(root == NULL) return;
        if(root->left == NULL && root->right == NULL) return ;
        flat(root->left);
        flat(root->right);
        if(root->left != NULL) {
            TreeNode* newRight = root->left;
            TreeNode* pRight = root->left;
            root->left = NULL;
            TreeNode* oldRight = root->right;
```

```
            while(pRight->right != NULL) {
                pRight = pRight->right;
            }
            pRight->right = oldRight;
            root->right = newRight;
        }
        return ;
    }
    void flatten(TreeNode* root) {
        if(root == NULL) return ;
        flat(root);
        return ;
    }
};
```

代码 迭代:

```
/**
 * 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 flatten(TreeNode* root) {
        if(root == NULL) return ;
        stack<TreeNode*> s;
        s.push(root);
        while(!s.empty()) {
            TreeNode* p = s.top();
            s.pop();
            if(p->right) s.push(p->right);
            if(p->left) s.push(p->left);
            if(!s.empty()) {
                p->right = s.top();
            }
            else {
                p->right = NULL;
            }
            p->left = NULL;
        }
        return;
    }
};
```