# 题目

二叉树数据结构TreeNode可用来表示单向链表（其中left置空，right为下一个链表节点）。实现一个方法，把二叉搜索树转换为单向链表，要求依然符合二叉搜索树的性质，转换操作应是原址的，也就是在原始的二叉搜索树上直接修改。

返回转换后的单向链表的头节点。

注意：本题相对原题稍作改动

**示例：**

输入： [4,2,5,1,3,null,6,0]

输出： [0,null,1,null,2,null,3,null,4,null,5,null,6]

**提示：**

节点数量不会超过 100000。

# 分析

## 方法一：中序遍历

**思路：**

先定义一个节点ans用来返回整棵树，定义另一个节点cur=ans，用来表示正在遍历的节点的上一个节点。

进行中序遍历，每遍历到一个节点时，就将它的左孩子设为NULL，然后将它身作为上一个节点的右孩子。

**代码：**

class Solution {

public:

TreeNode \*ans = new TreeNode(0),\*cur=ans;

void inOrder(TreeNode\* node)

{

if(node==NULL) return ;

inOrder(node->left);

node->left=NULL; //将该节点的左孩子设为NULL

cur->right=node; //将该节点赋给上一个节点的右孩子

cur=node; //更新cur

inOrder(node->right);

}

TreeNode\* convertBiNode(TreeNode\* root) {

inOrder(root);

return ans->right;

}

};