

时间限制：C/C++ 1秒，其他语言2秒 空间限制：C/C++ 64M，其他语言128M

热度指数：526422

本题知识点： 分治

 算法知识视频讲解

题目描述

输入一棵二叉搜索树，将该二叉搜索树转换成一个排序的双向链表。要求不能创建任何新的结点，只能调整树中结点指针的指向。

说明：本题目包含复杂数据结构TreeNode，[点此查看相关信息](#)

```
/*
struct TreeNode {
    int val;
    struct TreeNode *left;
    struct TreeNode *right;
    TreeNode(int x) :
        val(x), left(NULL), right(NULL) {
    }
};*/
class Solution {
public:
    void travel(TreeNode* root,TreeNode*&prev)//为了保证全局用一个 prev，（递归过程中
    prev 不是同一个值）采用指针引用
    {
        if(root)
        {
            //处理左子树
            travel(root->left,prev);
            //处理当前节点
            root->left=prev;
            if(prev)
            {
                prev->right=root;
            }
            prev=root;
            //处理右子树
            travel(root->right,prev);
        }
    }
}
```

```
TreeNode* Convert(TreeNode* pRootOfTree){  
    //中序链接 （搜索二叉树中序有序）  
    TreeNode* prev=nullptr;  
    travel(pRootOfTree,prev);  
    TreeNode* cur=pRootOfTree;  
    while(cur&&cur->left)  
    {  
        cur=cur->left;  
    }  
    return cur;  
}
```

```
};
```