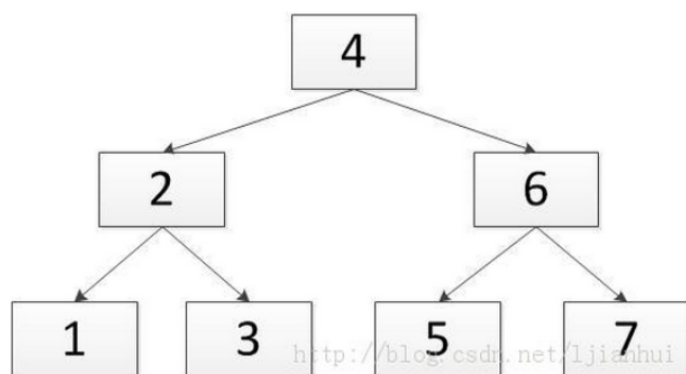


输入一颗二叉搜索树，将该二叉搜索树转换成一个排序的双向链表。要求不能创建任何新的节点，只能调整树中节点指针的指向。

**题目：**

输入一棵二叉搜索树，将该二叉搜索树转换为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。如下图：



若采用中序遍历，其遍历顺序为1-2-3-4-5-6-7，通过适当的指针变换操作，可变成的双向有序链表如下：



```
public class B50 {  
  
    public class TreeNode {  
        int val;  
        TreeNode left;  
        TreeNode right;  
        public TreeNode(int val) {  
            this.val = val;  
        }  
    }  
}
```

```
public TreeNode ConvertWithStack(TreeNode pRootOfTree) {
    if(pRootOfTree == null)
        return null;
    //如果根节点为null返回null, 否则当前节点cur指向根节点pRootOfTree
    TreeNode cur = pRootOfTree;
    TreeNode pre = null;
    //pre用于指向当前节点的前一个节点, 开始时为null
    Stack<TreeNode> stack = new Stack<B36.TreeNode>();
    while (!stack.isEmpty() || cur != null) {
        //将左子树上的节点入栈
        while (cur != null) {
            stack.push(cur);
            cur = cur.left;
        }
        //如果已经到了树的最左边, 则将栈中的节点出栈。
        cur = stack.pop();
        if(pre == null) {
            //刚开始的时候pre==null, 将最左边的节点赋值给pRootOfTree。当前节点赋值给pre
            pRootOfTree = cur;
            pre = cur;
        } else {
            pre.right = cur;
            cur.left = pre;
            pre = cur;
        }
        //前一个节点的右指针指向cur当前节点, 当前节点的左指针指向前一个节点pre
        cur = cur.right;
    }
    return pRootOfTree;
}
```