



链滴

二叉搜索树与双向链表

作者: [yudake](#)

原文链接: <https://ld246.com/article/1518856113652>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

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，只调整树中结点指针的指向。

解题思路

利用二叉树的中序遍历：

- 首先一直读取左链表，压入栈。
- 读取栈内最上面的值，此结点即为此时的最小值。
- 本结点的左链为上一结点，上一结点的右链为本结点。
- 检查本结点是否有右子树
 - 如果有右子树，那么对右子树重复本循环。
 - 如果没有右子树，那么回到读取栈值。

代码

```
import java.util.Stack;
public class Solution {
    public TreeNode Convert(TreeNode pRootOfTree) {
        if (pRootOfTree == null)
            return pRootOfTree;
        Stack<TreeNode> stack = new Stack<>();
        TreeNode pre = null;
        TreeNode head = null;
        while (!stack.isEmpty() || pRootOfTree != null) {
            while (pRootOfTree != null) {
                stack.push(pRootOfTree);
                pRootOfTree = pRootOfTree.left;
            }
            pRootOfTree = stack.pop();
            pRootOfTree.left = pre;
            if (pre != null)
                pre.right = pRootOfTree;
            if (head == null)
                head = pRootOfTree;
            pre = pRootOfTree;
            pRootOfTree = pRootOfTree.right;
        }
        return head;
    }
}
```