# 8. 二叉树的下一个结点

## 题目链接

[牛客网](https://www.nowcoder.com/practice/9023a0c988684a53960365b889ceaf5e?tpId=13&tqId=11210&tPage=1&rp=1&ru=/ta/coding-interviews&qru=/ta/coding-interviews/question-ranking&from=cyc_github)

## 题目描述

给定一个二叉树和其中的一个结点，请找出中序遍历顺序的下一个结点并且返回 。注意，树中的结点不仅包含左右子结点，同时包含指向父结点的指针。

public class TreeLinkNode {

 int val;
 TreeLinkNode left = null;
 TreeLinkNode right = null;
 TreeLinkNode next = null; // 指向父结点的指针

 TreeLinkNode(int val) {
 this.val = val;
 }
}

## 解题思路

我们先来回顾一下中序遍历的过程：先遍历树的左子树，再遍历根节点，最后再遍历右子树。所以最左节点是中序遍历的第一个节点。

void traverse(TreeNode root) {
 if (root == null) return;
 traverse(root.left);
 visit(root);
 traverse(root.right);
}

① 如果一个节点的右子树不为空，那么该节点的下一个节点是右子树的最左节点；

② 否则，向上找第一个左链接指向的树包含该节点的祖先节点。

public TreeLinkNode GetNext(TreeLinkNode pNode) {
 if (pNode.right != null) {
 TreeLinkNode node = pNode.right;
 while (node.left != null)
 node = node.left;
 return node;
 } else {
 while (pNode.next != null) {
 TreeLinkNode parent = pNode.next;
 if (parent.left == pNode)
 return parent;
 pNode = pNode.next;
 }
 }
 return null;
}