**题目: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。**

**A.代码(我的):**

class Solution {

public:

TreeLinkNode\* GetNext(TreeLinkNode\* pNode){

if (pNode == NULL)return NULL;

if (pNode->right != NULL){ //左结点为空

pNode = pNode->right;

while (pNode->left!=NULL){

pNode = pNode->left;

}

return pNode;

}

else { //pNode的右结点为空

int flag = 1;

TreeLinkNode \*pre = pNode;

while (flag){

if (pNode->next == NULL)return NULL;

pNode = pNode->next;//首先回退到父结点

if (pNode->right == pre)pre = pNode;

else flag = 0;

}

return pNode;

}

}

};

**B(示例).**

class Solution {

public:

TreeLinkNode\* GetNext(TreeLinkNode\* pNode){

if (pNode == NULL)return NULL;

if (pNode->right != NULL){

pNode = pNode->right;

while (pNode->left != NULL)pNode = pNode->left;

return pNode;

}

**else{**

**while (pNode->next && pNode->next->right == pNode)pNode = pNode->next;**

**if (!pNode)return NULL;**

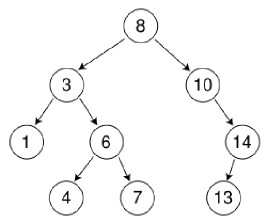
**else return pNode->next;**

**}**

}

};

**总结:A.考虑结点出现在7或者是14的情况。**



**B.**如果最右结点出现在左子树上，则if (pNode->right == pre)pre = pNode不会通过；

若最右结点出现在右字数上，则if (pNode->next == NULL)return NULL不会通过。

**C.**在树的递归长建议不需要flag这样的标志变量来判断是否为空，可直接使用TreeNode作为标志，比如**while (pNode->next && pNode->next->right == pNode)pNode = pNode->next;**