

# Chapter 05

谢胡睿 2400014151 数算（B）24-25春

书面作业：

- p168: 8, 9, 12, 13
- p169: 1, 4

8.

先根遍历为ABDEGCFHIJ

中根遍历为DBGEAHFIJC

Ans:

根为A

depth=1 A左：B（先根第二个）

depth=2 B左：D（中根第一个）

depth=2 B右：E（中根先G后E）

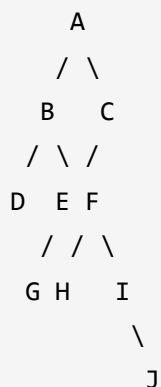
depth=3 E左：G

depth=1 A右：C（先根G后C+中根A后为右子树）

depth=2 C无右（中根最后C）C左：F

depth=3 F左：H F右：I

depth=4 I右：J



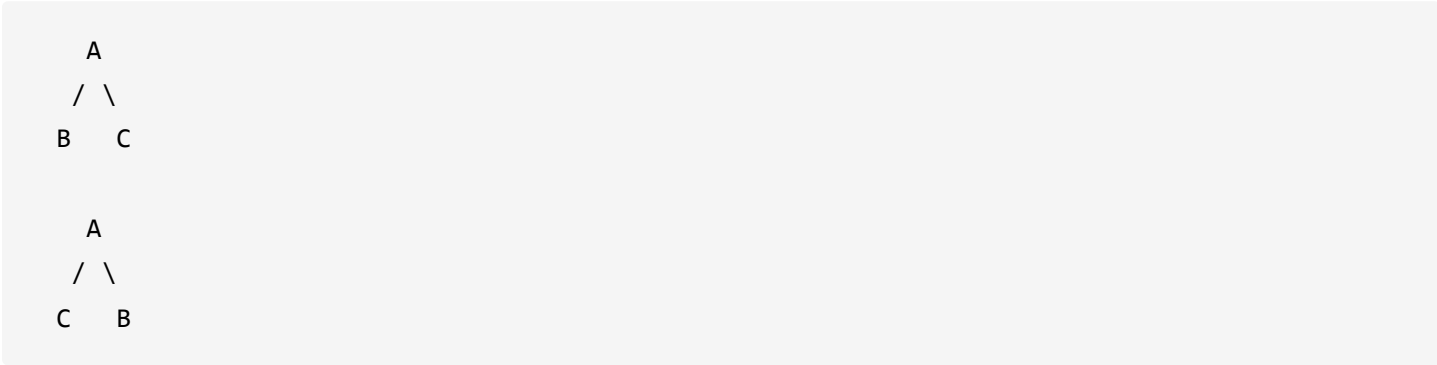
则后根遍历是：DGEBHJIFCA

9.

证明：  
若遍历为空，则树为空，不然：

- 1. 先根的第一个节点是根
- 2. 根在中根遍历中对应的位置，左侧是左子树，右侧是右子树
- 3. 递归即可建立整棵树

下证先根+后根不可  
先根：ABC  
后根：CBA



均符合，为反例

12.

父指针表示法：

index	info	parent
0	a	-1
1	b	0
2	c	0
3	d	0
4	e	1
5	f	2
6	g	2
7	h	2

## 子表示法

index	info	children
0	a	->b->c->d->nullptr
1	b	->e->nullptr
2	c	->f->g->h->nullptr
3	d	->nullptr
4	e	->nullptr
5	f	->nullptr
6	g	->nullptr
7	h	->nullptr

## 长子-兄弟表示法：

1st-child	info	next-sibling
b	a	nullptr
e	b	c
f	c	d
nullptr	d	nullptr
nullptr	e	nullptr
nullptr	f	g
nullptr	g	h
nullptr	h	nullptr

## 13.

显然：总共结点数为总度数+1

则结点数=  $\sum_{i=1}^m (i * n_i) + 1$

又  $N = n_0 + n_1 + n_2 + \dots + n_m = n_0 + \sum_{i=1}^m n_i$

则  $n_0 = 1 + \sum_{i=1}^m (i * n_i) - \sum_{i=1}^m n_i = 1 + \sum_{i=1}^m ((i - 1) * n_i)$

# 算法题

## 1.

```
struct Node{
    int index;
    Node* left;
    Node* right;
};
int countnumbers(Node* root){
    int ans=0;
    if(root==nullptr){
        return 0;
    }
    if(root->left==nullptr&&root->right==nullptr){
        return 1;
    }
    return countnumbers(root->left)+countnumbers(root->right);
}
```

## 4.

```
struct ThreadedNode{
    int data;
    ThreadedNode* left;
    ThreadedNode* right;
    bool leftThread;
    bool rightThread;
    ThreadedNode* parent;
};

ThreadedNode* preOrderSuccessor(ThreadedNode* node) {
    if (node==nullptr) return nullptr;
    if (!node->leftThread) return node->left; //有左子则左子根为前驱
    if (!node->rightThread) return node->right; //无左有右则右子根为前驱
    //向上找
    ThreadedNode* p=node->parent;
    ThreadedNode* current=node;
    while (p!=nullptr&&current==p->right) {
        current=p;
        p=p->parent;
    }
    return p;
}
```