**第六章 树**

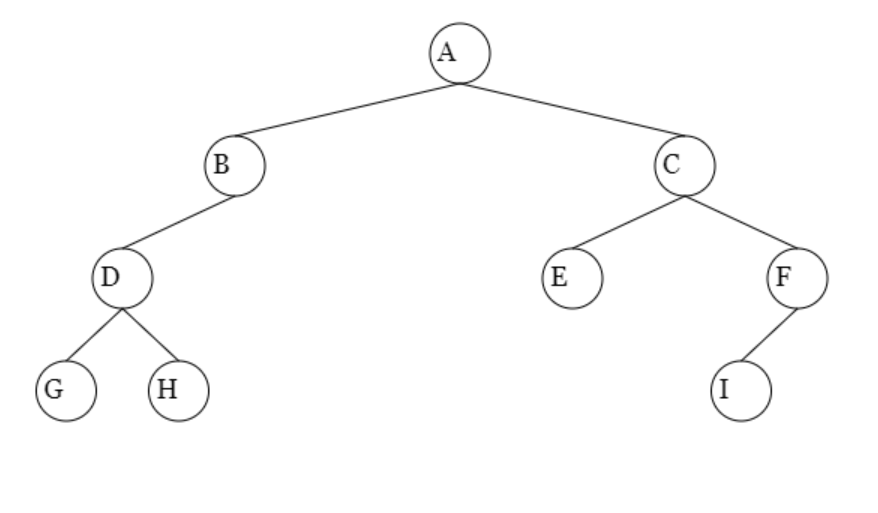
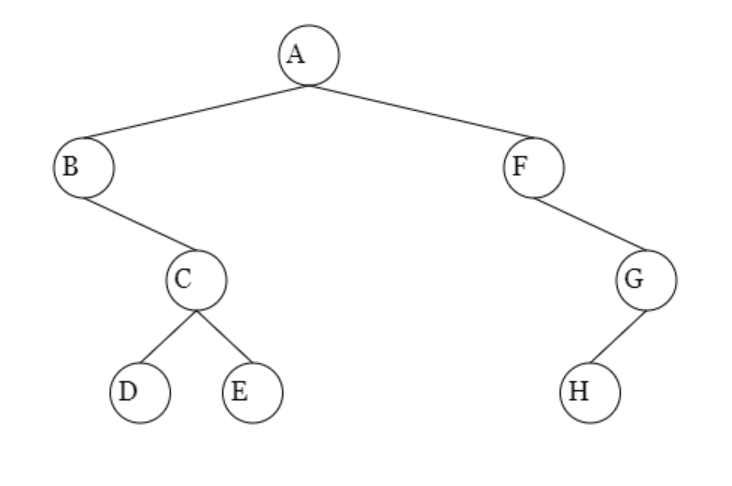
1. 如在内存中存放一个完全二叉树，在树上只进行下面两个操作：

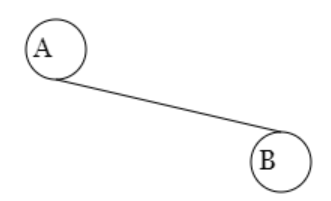
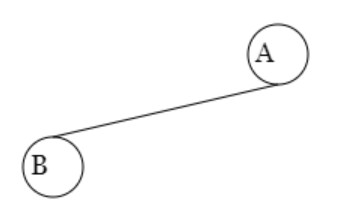
（1）寻找某个结点双亲 （2）寻找某个结点的儿子；

请问应该用何种结构来存储该二叉树？

数组，完全二叉树不怕空间浪费

2． 若二叉树中个结点的值均不相同，则由二叉树的前序序列和中序序列，或由其后序序列的中序列均能惟一地确定一棵二叉树，但由前序序列和后序序列却不一定能惟一地确定一棵二叉树。

1. 已知一棵二叉树的前序序列和中序序列分别为ABDGHCEFI和GDHBAECIF，请画出此二叉树。
2. 已知一棵二叉树的中序序列和后序序列分别为BDCEAFHG和DECBHGFA，请画出此二叉树。
3. 已知两棵二叉树前序序列和后序序列均为AB和BA，请画出这两棵不同的二叉树。



3. 假设用于通信的电子由字符集{a,b,c,d,e,f,g,h}中的字母构成，这8个字母在电文中出现的概率分别为{0.07，0.19，0.02，0.06，0.32，0.03，0.21，0.10}

1. 为这8个字母设计哈夫曼编码。

200

47 53

28 19 21 32

11 17

5 6 7 10

2 3

01

10

11

0001

0010

0011

00000

00001

1. 若用三位二进制数（0~7）对这个8个字母进行等长编码，则哈夫曼编码的平均码长是等长编码的百分之几？它使电文总长平均压缩多少？

(2\*(19+21+32)+4\*(6+7+10)+5\*(2+3))=144+92+25=261

261/300=87%

压缩13%

4.分别给出下图所示二叉树的先根、中根和后根序列。

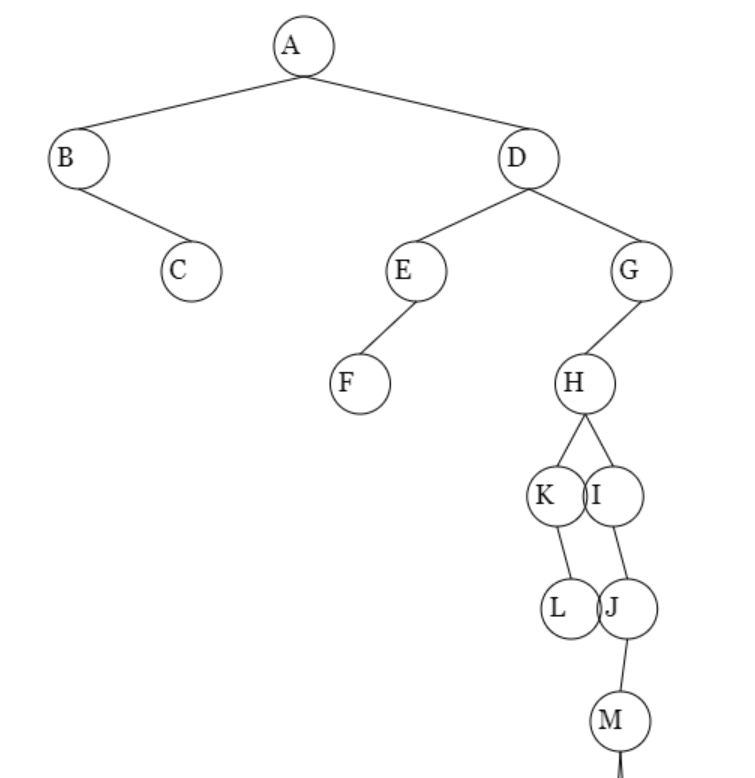


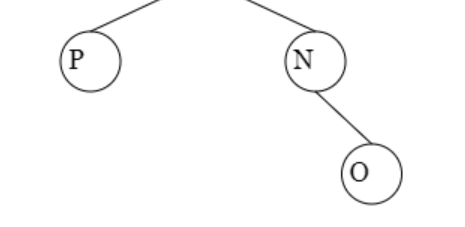
ABCDEF

BCDAFE

DCBFEA

5. 将下列由三棵树组成的森林转换为二叉树。（只要求给出转换结果）





6. 有一二叉链表，试编写按层次顺序遍历二叉树的算法。

void levelOrder(BTNode\* BT) {

SqQueue\* q; // 定义队列

initQueue(&q); // 初始化队列

if (BT != NULL) { // 根节点指针进队列

enQueue(q, BT);

}

// 一层一层的把节点存入队列，当没有孩子节点时就不再循环

while (!emptyQueue(q)) { // 队不为空循环

deQueue(q, &BT); // 出队时的节点

printf("%c", BT->data); // 输出节点存储的值

if (BT->lchild != NULL) { // 有左孩子时将该节点进队列

enQueue(q, BT->lchild);

}

if (BT->rchild != NULL) { // 有右孩子时将该节点进队列

enQueue(q, BT->rchild);

}

}

}

7. 已知二叉树按照二叉链表方式存储， 利用栈的基本操作写出先序遍历非递归形式的算法。

p＝b;

while (栈不空或者p!=NULL)

{

while (p!=NULL)

{ 访问p所指结点;将p进栈；

p=p->lchild

}

//以下考虑栈顶结点

if (栈不空)

{ 出栈p；

　　 　p＝p->rchild;

}

}

1. 已知完全二叉树的第9层有240个结点，则整个完全二叉树有多少个结点？有多少个叶子结点？

第九层满是2\*\*8=256个结点，所以一共九层

一共2\*\*9-1-256+240=495个结点

叶子结点240+(256-240)/2=248