



厦门大学《数据结构》期末试题·答案

考试日期：2006·1 (B)

信息学院自律督导部



一、 (本题 15 分) 试设计一个带头结点的单链表，然后写一个算法将该单链表逆转，要求利用原表结点空间，不允许申请和使用新的结点空间。

解：

方法一：建立一个新的单链表，其中的结点从原表得来，即每个原表中得到一个结点，就要将此结点插入新链表中。由于要将表逆转，原表的头结点成为新链表的头结点，每次从原表中得到一个结点，此结点插在头结点之后，作为新链表的第一个结点。

```
void InverLinkedList( LinkList &L){
    LNode *p, *s;
    P=L->next;
    L->next=NULL;
    while (p) {
        s=p;          //p 为待逆置链表头指针
        p=p->next;    //从 p 所指链表中删除第一个结点
        s->next=L->next;
        L->next=s;     //将 s 结点插入到逆置表的表头
    }
}
```

方法二：在遍历原表的时候将各结点的指针逆转，从原表的第一个结点开始，表头结点的指针在最后修改成指向原表的最后一个结点。

```
void InvertLinkedList( LinkList &L) {
    LNode *p, *q;
    S=L->next;
    if (s) {
        q=NULL;
        p=s;
        while (p) {
            p=p->next;
            s->next=q;
            q=s;
            s=p;
        }
        L->next=q;
    }
}
```

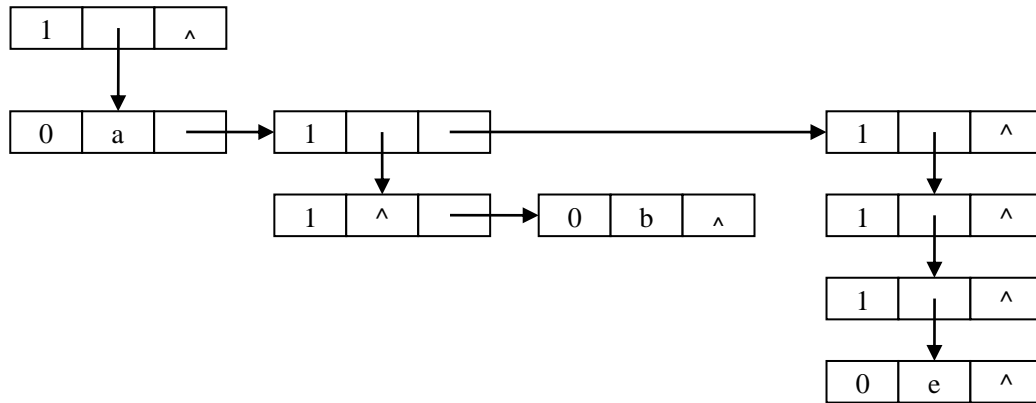
二、 (本题 10 分) 给定广义表 (a, ((), b), (((e))))，完成下列要求：

- 1) 给出广义表的数据结构；
- 2) 画出该广义表的存储结构图；
- 3) 利用取表头和表尾的操作分离出原子 e (给出 GetHead、GetTail 的操作序列)。

[解答]

1) (见课本 P109~110)

2)



3) 令给定广义表为 L, 则 $\text{GetHead}(\text{GetHead}(\text{GetHead}(\text{GetHead}(\text{GetTail}(\text{Gettail}(L)))))$

三、 (本题 10 分) 一棵二叉树的先序、中序和后序序列分别如下, 其中有一部分未显示出来, 试求出空格处的内容, 并画出该二叉树。

先序序列: B F ICEH G;

中序序列: D KFIA EJC ;

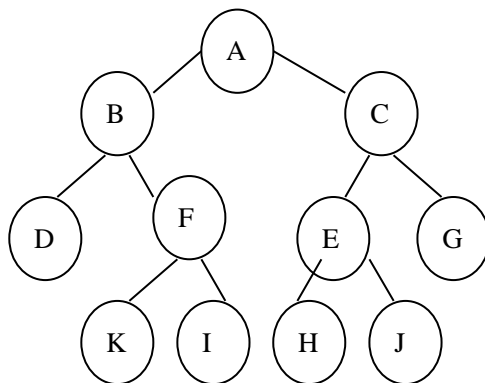
后序序列: K FBHJ G A。

解答: 先序序列为 ABDFKICEHJG;

中序序列为 DBKFIAHEJCG;

后序序列为 DKIFBHJEGCA。

二叉树如下:



四、 (本题 20 分) 设一棵二叉树以二叉链表表示, 试编写一算法统计二叉树的宽度, 即

在二叉树的各层上，求出具有结点数最多的那一层上的结点总数。

```
typedef struct BiTNode{
    TElemType data;
    Struct BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
```

[解答 1——递归]

```
LevelNumber(BiTree T, int NodeNum[], int level) {
// 求以*T 为根的子树中各层的宽度，存放在 NodeNum[]中，level 为*T 所在层次号。
    if (T!=NULL) {
        NodeNum[level]++;
        LevelNumber(T->lchild, NodeNum, h+1);
        LevelNumber(T->rchild, NodeNum, h+1);
    }
}

int BiTreeWidth(BiTree T) {
    for (i=0;i<=MAXLEVEL; i++) NodeNum[i]=0;
    LevelNumber(T, NodeNum, 0);
    wid=NodeNum[0];
    for (i=1;i<=MAXLEVEL; i++)
        if (wid<NodeNum[i]) wid=NodeNum[i];
    return wid;
}
```

[解答 2——按层序遍历]

```
typedef struct QElem {
    BiTree T;
    int Level;
} QElem, *QElemPtr;

int BiTreeWidth(BiTree *T) {
    InitQueue(Q);
    q=(QElemPtr) malloc(sizeof(QElem));
    q->T = T;
    q->level = 0;
    maxwid=wid=l=0;
    EnQueue(Q, q);
    while (!QueueEmpty(Q)) {
        DeQueue(Q, p);
        if (p->level==l) wid++;
        else {
            if (wid>maxwid) maxwid=wid;
            l=p->level; //或者 l++;
            wid=1;
        }
        q=(QElemPtr) malloc(sizeof(QElem));
        q->T = p->lchild;
        q->level = l+1;
    }
}
```

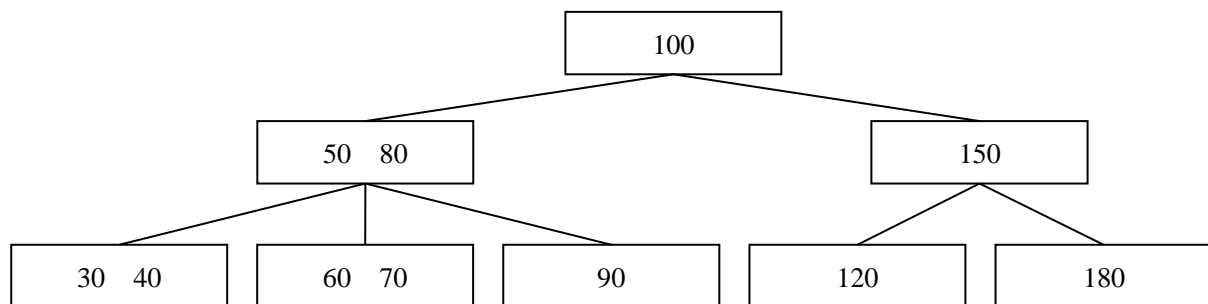
```

EnQueue(Q, q);

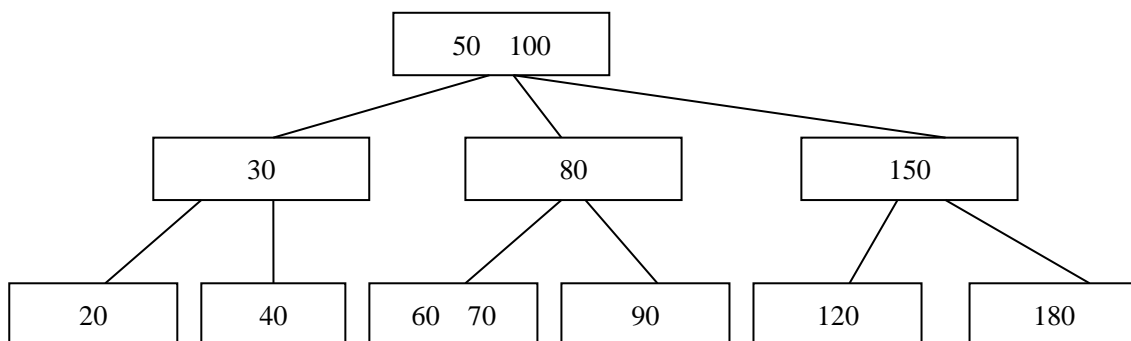
q=(QelemPtr) malloc(sizeof(QElem));
q->T = p->rchild;
q->level = l+1;
EnQueue(Q, q);
free(p);
}
DestroyQueue(Q);
Return maxwid;
}

```

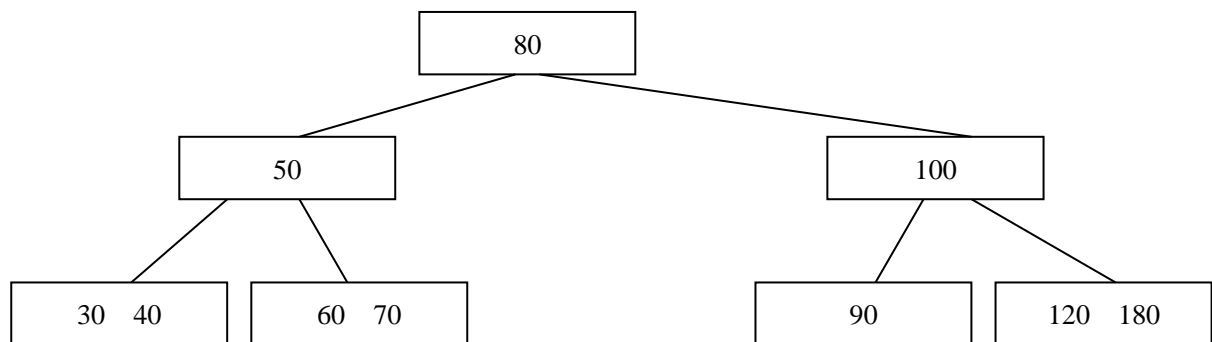
五、（本题 10 分）设有 3 阶 B-树，如下图所示，分别画出在该树插入关键字 20 和在原树删除关键字 150 得到的 B-树。



解：插入 20 后的 B-树为：



删除 150 后的 B-树为：



六、（本题 10 分）设待排序的表有 8 个记录，其关键字分别为：18，2，20，34，12，32，6，16。写出用 2-路归并排序的每趟结果。2-路归并排序算法是否是稳定的？

解答：

	(18)	(2)	(20)	(34)	(12)	(32)	(6)	(16)
一趟归并后	(2 18)	(20 34)	(12 32)	(6 16)				
二趟归并后	(2 18 20 34)	(6 12 16 32)						
三趟归并后	(2 6 12 16 18 20 32 34)							

2-路归并排序是稳定的。

七、（本题 20 分）若有大写字母、小写字母和数字组成的集合存放在一维数组中，请编写一个时间复杂度为 $O(n)$ 的算法，使得数组中的字符按大写字母、数字、小写字母的顺序排列，且辅助空间为 $O(1)$ 。

[提示] 本题只要求对字符按大写字母、数字、小写字母三种分类顺序排列，对同类字符之间的排列顺序并无特定要求。

[解答]

算法思想：

- 1) 第 1 趟：顺序扫描数组中的各个元素，通过交换将大写字母调整到数组的开头部分；
- 2) 第 2 趟：顺序扫描数组中的除大写字母外的部分，通过交换将小写字母调整到数组的尾部；

具体算法（略）

建议删除第八题，然后在调整、增加文件内容的试题之后，使用 B 卷考试。

八、（本题 5 分）请谈谈学习《数据结构》课程的心得体会，并以某个算法为例谈谈对该算法的理解。