

## 2002 硕士学位研究生入学试题

### 软件基础数据结构部分参考答案

#### 试题一、填空题

1. 分别采用堆排序、快速排序、插入排序和归并排序对初始状态为递增序列的表按递增顺序排序, 问最省时间的是 插入排序 算法, 最费时间的是 快速排序 算法。
2. 对下列给出的两种遍历序列判断是否可以唯一确定一棵二叉树, 用  $\checkmark$  表示可以。

|      | 先序遍历         | 中序遍历         | 后序遍历         | 层次遍历         |
|------|--------------|--------------|--------------|--------------|
| 先序遍历 |              | $\checkmark$ |              |              |
| 中序遍历 | $\checkmark$ |              | $\checkmark$ | $\checkmark$ |
| 后序遍历 |              | $\checkmark$ |              |              |
| 层次遍历 |              | $\checkmark$ |              |              |

3. 递归过程实现时使用的数据结构是 栈, 层次遍历二叉树时使用的数据结构是 队列。
4. 有向图用邻接矩阵表示, 删除所有从第  $i$  个结点出发的边的方法是 将第  $i$  行全置为 0。
5. 在数据结构中, 从逻辑上可以把数据结构分为 线性 结构和 非线性 结构。
6. 有  $n(n>0)$  个结点的  $d$  度树, 若用  $d$  个链域的多重链表表示, 则只有  $n-1$  个非空链表。
7. 已知广义表  $A = (a, b)$ ,  $B = (A, A)$ ,  $C = (a, (b, A), B)$ ,  
 $\text{tail}(\text{head}(\text{tail}(C)))$  的结果是  $(A)$ 。

#### 试题二、简答题

1. 设 HT 为哈夫曼树, 叶结点  $a$  的路径长度  $L_a$  比叶结点  $b$  的路径长度  $L_b$  长, 试证明: 叶结点  $b$  的权值  $W_b$  不小于叶结点  $a$  的权值  $W_a$ 。

证明: 反证法: 设  $W_b < W_a$

HT 的带权路径长度  $WPL = W_a \cdot L_a + W_b \cdot L_b + \text{Others}$

Others 为除  $a, b$  外其它叶结点的权值与路径长度乘积之和。

现交换叶结点  $a$  和  $b$

则新树 NewT 的带权路径长度  $WPL' = W_a \cdot L_b + W_b \cdot L_a + \text{Others}$

$WPL - WPL' = W_a \cdot L_a + W_b \cdot L_b - (W_a \cdot L_b + W_b \cdot L_a)$

$= W_a \cdot (L_a - L_b) - W_b \cdot (L_a - L_b)$

$= (W_a - W_b) \cdot (L_a - L_b) > 0$

与 HT 为哈夫曼树矛盾。

故  $W_b \geq W_a$

1. 简述哈希表查找方法。

哈希表查找方法不是通过关键字比较, 而是通过对关键字进行某种计算, 来实现关键字到存储地址的转换。

3. 对于  $n$  个顶点的无向图, 采用邻接矩阵表示, 回答下列问题:

(1) 如何求得图中的边数?

(2) 如何判断任意顶点  $i$  和  $j$  是否有边相连?

(3) 如何求任意一个顶点的度?

答: (1) 图中的边数等于邻接矩阵的非零元素个数之和的一半。

(2)  $i$  和  $j$  有边相连的条件是  $A[i,j] \neq 0$  且  $A[j,i] \neq 0$ 。

(3) 顶点  $j$  的度为第  $j$  行中非零元素的个数。

### 试题三、算法题参考答案

1. 修改下面层次遍历二叉树算法使其能输出按层次遍历顺序所访问的第一个和最后一个叶结点。

PROC levelTrav (t: bitre);

{t 为用二叉链表存储的二叉树根指针, 二叉链表的结点结构为:

|        |      |        |
|--------|------|--------|
| lchild | data | rchild |
|--------|------|--------|

IF t  $\neq$  NIL THEN

[ iniqueue(Q); {初始化队列

enqueue(Q, t); {入队列

tag:=true;

WHILE NOT EMPTY(Q) DO {队列非空

[ p:=dequeue(Q); {出队列

IF tag THEN IF p  $\uparrow$ .lchild = NIL AND p  $\uparrow$ .rchild = NIL  
THEN [ write(p  $\uparrow$ .data 是第一个叶结点

tag:=false ]

IF p  $\uparrow$ .lchild  $\neq$  NIL THEN enqueue(Q, p  $\uparrow$ .lchild);

IF p  $\uparrow$ .rchild  $\neq$  NIL THEN enqueue(Q, p  $\uparrow$ .rchild)

];

write(p  $\uparrow$ .data 是最后一个叶结点

]

ENDP; { levelTrav

2.

PROC DELETE(L, x);

p:=L  $\uparrow$ .firstLink; q:=L; {置初值: p, q 为同步指针, q 为 p 的直接前驱

WHILE p  $\neq$  NIL DO

[ IF p  $\uparrow$ .data=x THEN

CASE

L=q: [ L  $\uparrow$ .firstLink:=p  $\uparrow$ .link; {第一个为 x

IF p  $\uparrow$ .link=NIL THEN L  $\uparrow$ .lastLink:=NIL {只有一个结点且为 x  
p:=p  $\uparrow$ .link];

L  $\uparrow$ .lastLink=p: [ L  $\uparrow$ .lastLink:=q;

q  $\uparrow$ .link:=p  $\uparrow$ .link; p:=p  $\uparrow$ .link ] {最后一个为 x

Others: [ q  $\uparrow$ .link:=p  $\uparrow$ .link; p:=p  $\uparrow$ .link ]

ENDC

ELSE [ q:=p; p:=p  $\uparrow$ .link ]

]

ENDP; {DELETE



## 参考答案

### 五、单项选择题（请选出正确的编号填入括号中，每小题 1 分，共 10 分）

- 1、②    2、②    3、②    4、④    5、③  
6、④    7、④    8、③    9、①    10、③

### 六、填空题（每小题 1 分，共 9 分）

- 1、CPU, I/O
- 2、资源数，等待资源的进程数
- 3、安全性，死锁
- 4、信号量机制
- 5、顺序性、封闭性、可再现性
- 6、信息维护，通信系统
- 7、透明性，健壮性
- 8、重入码，虚拟存储器
- 9、请求式调页，预先调页法

### 七、判断题（正确的在括号内打√，错误的在括号内打×，每小题 1 分，共 10 分，）

- 1、×    2、×    3、×    4、√    5、√  
6、×    7、×    8、√    9、√    10、√

### 八、问答题（每小题 7 分，共 21 分）

- 1、答：位示图需要 2000 字节的空间。当空闲表数目大于 400 个时，则空闲表大于位示图。
- 2、答：进程同步机制中的让权等待的概念是：当进程不能获得资源时，放弃处理机，避免忙等。采用让权等待，主要是为了更有效的发挥 CPU 的性能，提高系统的吞吐量。
- 3、①答案：

| 进程 | Allocation |   | Max |   | Need |   | Available |   |
|----|------------|---|-----|---|------|---|-----------|---|
|    | A          | B | A   | B | A    | B | A         | B |
| P0 | 2          | 0 | 2   | 4 | 0    | 4 | 2         | 7 |
| P1 | 3          | 2 | 10  | 2 | 7    | 0 |           |   |
| P2 | 1          | 4 | 5   | 4 | 4    | 0 |           |   |

|    |   |   |   |   |   |   |  |  |
|----|---|---|---|---|---|---|--|--|
| P3 | 2 | 1 | 3 | 1 | 1 | 0 |  |  |
| P4 | 0 | 0 | 4 | 2 | 4 | 2 |  |  |

②安全序列：P0, P2, P3, P4, P1