

## 中科院计算机技术研究所 1998 年硕士生入学试题 数据结构和程序设计

要求: 算法题目写注解

### 一. 填空 (15 分, 每空一分)

1. 用循环链表表示的队列长度为  $n$ , 若只设头指针, 则出队和入队的时间复杂度分别是\_\_和\_\_  
\_&#59; 若只设尾指针, 则出队和入队的时间复杂度分别是\_\_和\_\_.
2. 设广义表  $L = ((), ())$ , 则  $\text{head}(L)$  是\_\_&#59;  $\text{tail}(L)$  是\_\_&#59;  $L$  的长度是\_\_&#59; 深度是\_\_.
3. 深度为  $h$  的完全二\*树至少有\_\_个结点&#59; 至多有\_\_个结点&#59;  $h$  和结点总数  $n$  之间的关系是\_\_.
4. 在  $n$  个记录的有序顺序表中进行折半查找, 最大的比较次数是\_\_.
5. 在一棵  $m$  阶 B+树中, 若在某结点中插入一个新关键字而引起该结点分裂, 则此结点中原有的关键字的个数是\_\_.
6.  $n$  个顶点的连通图用邻接矩阵表示时, 该矩阵至少有\_\_个非零元素.

### 二. 请在下列各题中选择一个正确的答案 (20 分, 每题 2 分)

1. 算法的时间复杂度取决于
  - a. 问题的规模
  - b. 待处理数据的初态
  - c. both a and b
2. 消除递归不一定需要使用栈, 此说法
  - a. true
  - b. false
3. 假定有  $k$  个关键字互为同义词, 若用线性探测法把这  $k$  个关键字存入散列表中, 至少要进行多少次探测?
  - a.  $k-1$
  - b.  $k$
  - c.  $k+1$
  - d.  $k(k+1)/2$
4. 若需要在  $O(n \log_2(n))$  的时间内完成对数组的排序, 且要求排序是稳定的, 则可选的排序方法是:
  - a. 快速排序
  - b. 堆排序
  - c. 归并排序
  - d. 直接插入排序
5. 用 ISAM 和 VSAM 组织文件属于:
  - a. 顺序文件
  - b. 索引文件
  - c. 散列文件

6. 若一个有向图的邻接矩阵中, 主对角线以下的元素均为零, 则该图的拓扑有序序列

- a. 存在
- b. 不存在

7. 将两个各有  $n$  个元素的有序表归并成一个有序表, 其最少的比较次数是

- a.  $n$
- b.  $2n-1$
- c.  $2n$
- d.  $n-1$

8. 下述二\*树中, 那一种满足性质: 从任意结点出发到根的路径上所经过的结点序列按其关键字有序:

- a. 二\*排序树
- b. 哈夫曼树
- c. AVL 树
- d. 堆

9. 以知待排序的  $n$  个元素可分为  $n/k$  个组, 每个组包含  $k$  个元素, 且任一组内的个元素均分别大于前一组内的所有元素和小于后一组内的所有元素, 若采用基于比较的排序, 其时间下限应为:

- a.  $O(k \log_2(k))$
- b.  $O(k \log_2(n))$
- c.  $O(n \log_2(k))$
- d.  $O(n \log_2(n))$

10. 在叶子数目和权值相同的所有二\*树中, 最优二\*树定是完全二\*树, 该说法:

- a. 正确
- b. 错误

三. 设二\*排序树  $T$  中各结点关键字互不相同,  $x^{\wedge}$  是  $T$  的叶子,  $y^{\wedge}$  是  $x^{\wedge}$  的双亲. 证明  $y^{\wedge}.key$  是  $T$  中大于  $x^{\wedge}.key$  的所有关键字中的最小者, 或是小于  $x^{\wedge}.key$  的所有关键字的最大者. (10 分)

四. (共 15 分) 设数组  $A$  的长度为  $2N$ , 前  $N$  个元素  $A[1..N]$  递减有序, 后  $N$  个元素  $A[N+1..2N]$  递增有序, 且  $2N$  是 2 的整数次幂, 即  $k=\log_2(2N)$  为整数. 例如  $A[1..8]=[90, 85, 50, 10, 30, 65, 80, 100]$  满足上述要求, 这里  $N=4, k=3$ ,  $A$  的前 4 个元素和后 4 个元素分别递减和递增有序. 用次例调用如下的 Demo 过程, 并要求:

(1). 给出 for 循环中每次执行 PerfectShuffle( $A, N$ ) 和 CompareExchange( $A, N$ ) 的结果. (10 分)

(2) 解释 Demo 的功能. (2 分)

(3) 给出 Demo 的时间复杂度. (3 分)

```
Procedure PerfectShuffle (Var A:arraytype; N:integer) {  
  i:=1; j:=1;  
  while i<=N do {  
    B[j]:=A[i];  
    B[j+1]:=A[i+N];
```

```

i:=i+1;
j:=j+2;
}
A[1..2N]:=B[1..2N]; //B copy to A
}

Procedure CompareExchange (Var A: arraytype; N: integer) {
j:=1;
while j<2N do {
if A[j]>A[j+1] then
A[j]<->A[j+1]; //exchange A[j] and A[j+1]
j:=j+2;
}
}

Procedure Demo (Var A: arraytype; N: integer) {
//the length of A is 2N, k=log2(N) is integer
for i:=1 to log2(2N) do
{PerfectShuffle(A, N);
CompareExchange(A, N);
}
}

```

##### 五. (共 20 分)

(1). 设二\*排序中关键字由 1 至 1000 的整数构成, 现要检索关键字为 363 的结点, 下述关键字序列中那些可能是二\*排序树上搜索到的序列, 那些不可能是二\*排序树上搜索到的序列? (5 分)

- (a) 2, 252, 401, 393, 330, 344, 397, 363
- (b) 924, 220, 911, 244, 898, 258, 362, 363
- (c) 925, 202, 911, 240, 912, 245, 363
- (d) 2, 399, 387, 219, 266, 382, 381, 278, 363

(2). 通过对 (1) 的分析, 写一个算法判定给定的关键字序列 (假定关键字互不相同) 是否可能是二\*排序树的搜索序列. 若可能是返回真, 否则返回假. 可假定被判定的序列已存入数组中. (15 分)

六. (共 20 分) 图的 D-搜索类似于 BFS, 不同之处在于使用栈代替 BFS 中的队列, 入出队列的操作改为入出栈的操作. 即当一个顶点的所有邻接点被搜索后, 下一个搜索的出发点应该是最近入栈 (栈顶) 的顶点.

(1) 用邻接表做存储结构, 写一个 D-搜索算法 (15 分)

(2) 用 D-搜索方法搜索右图, 设初始出发点为 1, 写出顶点的访问次序和响应的生成树, 当从某顶点出发搜索他的邻接点是, 请按邻接点序号递增序搜索, 以使答案唯一. (5 分)