

中国科学院软件研究所

2001 年招收攻读硕士学位研究生入学试卷

试题名称: 程序设计

一. 单项选择题 (每空 2 分, 共 20 分)

- 下列函数中渐近时间复杂度最小的是_____。
(a) $T_1(n) = n \log_2 n + 5000n$ (b) $T_2(n) = n^2 - 8000n$
(c) $T_3(n) = n^{\log_2 2.1} - 6000n$ (d) $T_4(n) = 2n \log_2 n - 7000n$
- 线性表的静态链表存储结构与顺序存储结构相比优点是_____。
(a) 所有的操作算法实现简单 (b) 便于随机存取
(c) 便于插入和删除 (d) 便于利用零散的存储器空间
- 设栈的输入序列为 $1, 2, \dots, n$, 输出序列为 a_1, a_2, \dots, a_n , 若存在 $1 \leq k \leq n$ 使得 $a_k = n$, 则当 $k \leq i \leq n$ 时, a_i 为_____。
(a) $n-i+1$ (b) $n-(i-k)$ (c) 不确定
- 设高度为 h 的二叉树 (根的层数为 1) 上只有度为 0 和度为 2 的节点, 则此类二叉树中所包含的节点数至少为_____。
(a) $2h$ (b) $2h-1$ (c) $2h+1$ (d) $h+1$
- 设指针 p 指向线索树中的某个结点, 则查找 $*p$ 在某种次序下的前驱或后继不能获得加速的是_____。
(a) 前序线索树中查找 $*p$ 的前序后继
(b) 中序线索树中查找 $*p$ 的中序后继
(c) 中序线索树中查找 $*p$ 的中序前驱
(d) 后序线索树中查找 $*p$ 的后序后继
- 假定有 k 个关键字互为同义词, 若用线性探测法将这 k 个关键字存入散列表中, 至少需要进行_____次探测。
(a) $k-1$ (b) k (c) $k+1$ (d) $k(k+1)/2$
- 若待排序元素基本有序, 则下列排序中平均速度最快的排序是_____; 若要求辅助空间为 $O(1)$, 则平均速度最快的排序是_____; 若要求排序是稳定

的,且关键字为实数,则平均速度最快的排序是_____。

- (a) 直接插入排序 (b) 直接选择排序 (c) Shell 排序
(d) 冒泡排序 (e) 快速排序 (f) 堆排序
(g) 归并排序 (h) 基数排序

8. 对于多关键字而言,_____是一种方便而又高效的文件组织方式。

- (a) 顺序文件 (b) 索引文件 (c) 散列文件 (d) 倒排文件

二. 问答题 (共 25 分)

1. 设 $A[-2:6;-3:6]$ 是一个用行主序存储的二维数组,已知 $A[-2,-3]$ 的起始存储位置为 $loc(-2,-3)=1000$,每个数组元素占用 4 个存储单元,求:
(6 分)

- 1) $A[4,5]$ 的起始存储位置 $loc(4,5)$;
2) 起始存储位置为 1184 的数组元素的下标。

2. 给定二叉树的先序和后序遍历序列,能否重构出该二叉树?若能,试证明之,否则给出反例。(6 分)

3. 在含有 n 个关键字的 m 阶 B-树中进行查找,至多读盘多少次?完全平衡的二叉排序树的读盘次数大约比它大多少倍?(设两种树中的每个节点均是一个存储块)(8 分)

4. 用向量表示的循环队列的队首和队尾位置分别为 1 和 max_size ,试给出判断队列为空和为满的边界条件。(5 分)

三. 阅读程序题 (共 10 分)

1. 设 G 是一个有向无环图,试指出下述算法的功能,输出的序列是 G 的什么序列?(10 分)

```
void Demo( ALGraph G ){
    //G 是图的逆邻接表, 向量 outdegree 的各分量初值为 0。
    for ( i=0; i<G.NodeNum; i++ )
        for ( p=G.adjlist[i].firstedge; p; p=p->next )
            //扫描 i 的入边表
            outdegree[p->adjvex]++; //设 p->adjvex=j, 则将
            //<j, i>的起点 j 的出度加 1

    InitStack( &S ); // 设置空栈 S
    for( i=0 ; i<G.NodeNum ; i++ )
```

```

    if( outdegree[i]==0 )
        Push( &S , i ) ; //出度为 0 的顶点 i 入栈
while( !StackEmpty(S) ){ //栈 S 非空
    i = Pop( &Q ) ; // 出栈, 相当于删去顶点 i
    printf("%c", G.adjlist[i].vertex ); //输出顶点 i
    for ( p=G.adjlist[i].firstedge; p; p=p->next ) {
        //扫描 i 的入边表
        j= p->adjvex; //j 是 i 的入边<j, i>的起点
        outdegree[j]--; //j 的出度减 1, 即删去 i 的入边<j, i>
        if ( !outdegree[j] )
            Push( &S, j ); // 若 j 的出度为 0, 则令其入栈
    } //end for
} // end while
} // end Demo

```

四. 算法题 (每题 15 分, 共 45 分)

1. 试设计算法在 $O(n)$ 时间内将数组 $A[1..n]$ 划分为左右两个部分, 使得左边的所有元素值均为奇数, 右边的所有元素值均为偶数, 要求所使用的辅助存储空间大小为 $O(1)$.
2. 试写一递归算法, 从大到小输出二叉排序树中所有的值不小于 x 的关键字. 要求算法的时间为 $O(h+m)$, 其中 h 为树的高度, m 为输出的关键字个数.
3. 设 G 是以邻接表表示的无向图, v_0 是 G 中的一个顶点, k 是一个正的常数. 要求写一算法打印出图中所有与 v_0 有简单路径相通, 且路径长度小于等于 k 的所有顶点(不含 v_0), 路径长度由路径上的边数来定义.

一. 单选题

1 (a) 2. (c) 3. (c) 4. (b) 5 (d) 6 (d) 7 (a), (f), (g) 8 (d)

二. 简答题

1. 1) 1272 2) A[2,3]

2. 不行. 例如 $\begin{matrix} a & \\ / & \backslash \\ b & \end{matrix}$ $\begin{matrix} a & \\ / & \backslash \\ & b \end{matrix}$

3. B-树为 $O(\log_t n)$, $t = \lceil m/2 \rceil$

平衡二叉树为 $O(\log_2 n) \because \log_2 n = \log_t n \log_2 t$, 即是B-树的大约 $\log_2 t$ 倍.
即它的复杂度比B-树大约大 $\log_2 t$ 倍.

三. 由读程序题

前序遍历, 与后序遍历序列.

四. 算法题

1. void partition (int a[], int n) {

$i=1; j=n;$

while (i < j) do {

while (i < j and odd(a[i])) do $i=i+1;$

while (i < j and !odd(a[j])) do $j=j-1;$

if (i < j) {

a[i] 和 a[j] 交换; $i++; j--;$

}

}

2. 使用从右向左的遍历, 但只停在对小于 x 的元素的分支不遍历, 否则 $O(\log n + m)$ 时间复杂度不保证.

3. 采用广度优先搜索, 从 1 出发, 在每层结点前增加标记 "#", 另设一计数器 m.