

青岛大学 2006 年硕士研究生入学考试试题 (A 卷)

科目代码: 407

科目名称: 数据结构(共 4 页)

请考生写明题号, 将答案全部答在答题纸上, 答在试卷上无效

一、单项选择题 (本大题共 15 道小题, 每小题 2 分, 共 30 分)

1. 设有广义表 $D(a, b, D)$, 其深度为: 【 】
A. 3 B. 1 ☒ C. ∞ D. 不确定
2. 下面程序段中带下划线的语句的执行次数的数量级是: 【 】
i: =1; WHILE i<n DO i: =i*2;
A. $O(n)$ ☒ B. $O(\log_2 n)$
C. $O(n^2)$ D. $O(1)$
3. 若用一个大小为 8 的数组来实现循环队列, 且当前 rear 和 front 的值分别为 0 和 3, 当从队列中删除一个元素, 再加入两个元素后, rear 和 front 的值分别为: 【 】
A. 1 和 7 B. 2 和 6 C. 6 和 2 ☒ D. 7 和 1
4. 递归过程或函数调用时, 处理参数及返回地址, 要用一种称为 () 的数据结构。
A. 队列 B. 多维数组 ☒ C. 栈 D. 线性表
5. 表达式 $a*(b+c)-d$ 的后缀表达式是: 【 】
A. $abcd*+-$ ☒ B. $abc*+d-$ C. $abc**d-$ D. $-+*abcd$
6. 某二叉树的中序遍历序列为 CBEDHAGIJF, 后序遍历序列为 CEDBAJIGFAH, 则前序序列为 【 】
☒ A. HBCDEFGAIJ B. HBCDGAIJEF C. HCDEBAJIGF D. HCBDEFGAIJ
7. 假设以行序为主序存储二维数组 $array[0..99, 0..99]$, 设每个数据元素占 2 个存储单元, 基地址为 10, 则 $LOC[5, 5]$ 为: 【 】
A. 808 B. 818 C. 1010 ☒ D. 1020
8. 对稀疏矩阵进行压缩存储目的是 ()。
A. 便于进行矩阵运算 B. 便于输入和输出 ☒ C. 节省存储空间 D. 降低运算的时间复杂度
9. 下面关于串的叙述中, 哪一个是不正确的? 【 】
A. 串是字符的有限序列 ☒ B. 串是线性表
C. 模式匹配是串的一种重要运算 D. 串采用顺序存储, 不能采用链式存储
10. 已知一个完全有向图共有 $(n+1)*n$ 条弧, 其顶点数为: 【 】
A. n ☒ B. n+1 C. n-1 D. 以上都不对
11. 二分法查找只适用于查找顺序存储的有序表, 在此假定 N 为线性表中结点数, 且每次查找都是成功的, 查找一个关键字次数最多不超过 (最接近) 次数为: 【 】
A. N+1 ☒ B. $\log_2 N + 1$ C. $\log N / 2$ D. N/2
12. 在排序算法中, 那种排序方法所需要的辅助空间最大。 【 】
A. 简单排序 ☒ B. 快速排序 C. 堆排序 D. 归并排序。

13. 下述文件中适合于磁盘存储的是 ()。

【 】

A. 顺序文件 ☒ B. ISAM 文件 C. 散列文件 D. 多关键字文件

14. 对包含 n 个元素的哈希表进行查找时, 平均查找长度

【 】

A. 为 $O(\log_2 n)$ B. 为 $O(n)$ ☒ C. 不直接依赖于 n D. 以上都不对

15. 对于含有 n 个顶点 e 条边的无向连通图, 利用普里姆算法产生最小生成树时, 其时间复杂度为

【 】

☒ A. $O(n^2)$ B. $O(n \cdot e)$ C. $O(n \log_2 n)$ ☒ D. $O(e \log_2 e)$

二、填空题 (本大题共 10 小题, 每小题 2 分, 共 20 分)

1. 当线性表的元素总数基本稳定, 且很少进行插入和删除操作, 但要求以最快的速度存取线性表中的元素时, 应采用 顺序 存储结构。

2. 设有一个顺序栈 S , 元素 $s_1, s_2, s_3, s_4, s_5, s_6$ 依次进栈, 如果 6 个元素的出栈顺序为 $s_2, s_3, s_6, s_5, s_4, s_1$, 则顺序栈的容量至少应为 4。

3. 含有 n 个结点的二叉树, 分支为 2 的结点数为 n_2 , 分支为 1 的结点数为 $n - 2n_2 - 1$ 。

4. 在进行直接插入排序时, 其数据比较次数与数据的初始排列 有 关; 而在进行直接选择排序时, 其数据比较次数与数据的初始排列 无 关。

5. 有向图是否存在回路的判定方法有 拓扑排序 和 深度优先遍历 方法。

6. 已知广义表 $A = (((a, b), (c), (d, e)))$, $\text{head}(\text{tail}(\text{tail}(\text{head}(A))))$ 的结果是 (d, e) 。

7. 在顺序存储的完全二叉树中, 编号为 i 和 j 的两个结点处在同一层的条件是 $\lfloor \log_2 i \rfloor = \lfloor \log_2 j \rfloor$ 。

8. 已知一无向图 $G = (V, E)$, 其中 $V = \{a, b, c, d, e\}$ $E = \{(a, b), (a, d), (a, c), (d, c), (b, e)\}$ 现用某一种图遍历方法从顶点 a 开始遍历图, 得到的序列为 $abdce$, 则采用的是 层次 遍历方法。

9. 在森林转化为二叉树结构后的, 其森林的后根遍历方法对应二叉树的 中序 遍历方法。

10. 动态查找表和静态查找表的重要区别在于前者包含有 插入 和 删除 运算, 而后者不包含这两种运算。

三、判断题 (本大题共 5 道小题, 每小题 2 分, 共 10 分。答题时写“对”或“错”)

1. 在完全二叉树中, 若一个结点没有左孩子, 则必为叶子结点 (对)

2. 需要借助于一个栈来实现非递归 BFS 算法。 (对)

3. 在动态存储管理中, 伙伴系统中的伙伴是指任意两块大小相同、位置相邻的内存 (错)

4. 外部排序所需的总的时间是内部排序的时间总和。 (错)

5. 归并排序算法要比快速排序算法占用更多的辅助单元。 (错)

四、应用题 (共 6 道小题, 1-4 题 各 6 分, 5 题 11 分, 6 题 15 分, 共 50 分)

1. 给定一个关键字序列 $\{55, 31, 11, 37, 46, 73, 63, 02, 07\}$, 用画图方法表示出建立二叉排序平衡树的过程。

2. 使用一维数组 $\text{Array}[\text{MAX}]$ 来存储循环队列的元素, 如果不设置标志 flag 来区分头指针 (front) 和尾指针 (rear) 相等时, 队列的状态是“空”还是“满”如何判断?

- 有一个原始数列为 {48, 38, 65, 97, 76, 13, 27, 49}, 经过一趟排序后的结果为: {27, 38, 13, 48, 76, 97, 65, 49}, 是使用了何种排序方法? 用该方法完成排序, 写出每趟排序的结果。
- 给定一个关键字序列 12, 56, 34, 23, 98, 67, 100, 63, 要求哈希表的装填因子为 0.8, 使用 7 求余数方法构造哈希函数, 使用开放定址的二次探测再散列方法处理冲突, 将 7 个数据放入到哈希表中, 用语言描述哈希表构造和查找方法。
- 从程序设计的角度来理解并说明算法的五个特征 (有穷性、确定性、可行性、输入、输出) 和一个好的算法应该达到的目标 (正确性、可读性、健壮性、效率与低存储量需求), 在具体的应用中如何解决算法的效率和低存储量需求的矛盾。
- 对有五个结点 {A, B, C, D, E} 的图的邻接矩阵,

$$\begin{bmatrix} 0 & 80 & 30 & \infty & 10 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 60 & 0 & 20 & \infty \\ \infty & 10 & \infty & 0 & \infty \\ \infty & \infty & \infty & 70 & 0 \end{bmatrix}$$

- 画出逻辑图;
- 画出图的邻接表存储;
- 基于邻接矩阵写出图的深度、广度优先遍历序列;
- 计算图的关键路径。

五、算法设计题 (本大题共 5 道小题, 每小题 8 分, 共 40 分)

- 下面是求二叉树高度的类 C 写的递归算法试补充完整

二叉树的两指针域为 lchild 与 rchild, 算法中 p 为二叉树的根, lh 和 rh 分别为以 p 为根的二叉树的左子树和右子树的高, hi 为以 p 为根的二叉树的高, hi 最后返回。

```
height(p)
{
    if ((1) p != null)
    {
        if (p->lchild == null) lh = (2) 0; else lh = (3) height(p->lchild)
        if (p->rchild == null) rh = (4) 0; else rh = (5) height(p->rchild)
        if (lh > rh) hi = (6) lh; else hi = (7) rh;
    }
    else hi = (8) 0;
    return hi;
}
```

- 本题是用 C 语言编写的一个双向起泡的递增排序算法, 即相邻两遍向相反方向起泡。其中有几处空白, 试补充完整。

```
void Bubble_Sort2(int a[], int n) // 相邻两趟是反方向起泡的冒泡排序算法
{
```



```

low=0;high=n-1; //冒泡的上下界
change= 1;
while(low<high&& (1) change )
{
    change= (2) 0;
    for(i=low;i< (3) n/2;i++) //从上向下起泡
        if(a[i]>a[i+1])
        {
            a[i]<->a[i+1];
            (4) change change=1;
        }
        (5) low low=low+1;
    for(i=high;i> (6) n/2;i--) //从下向上起泡
        if(a[i]<a[i-1])
        {
            a[i]<->a[i-1];
            (7) change=1;
        }
        (8) high high=high-1;
}
} //while
} //Bubble_Sort2

```

3. 用插入排序方法写出建立一个按关键字递增的有头结点单向链表的函数
【说明】结构定义

```

struct ListNode {
    keytype    key;
    struct ListNode *next;
};

```

4. 应用栈的算法写出算法检查如下字符串 $\#a/(1*(3+x)/y-(a-b))+z\#$ 的括号匹配问题。
5. 已知一棵二叉树的前序序列和中序序列分别存放于一维数组 a 和数组 b 中，编写建立该二叉树的二叉链表的算法。

树的结点类型描述如下

```

struct Node {
    int    key;
    struct Node *llink,*rlink;} ;

```