

# 广东工业大学

## 2012 年攻读硕士学位研究生入学考试试题

考试科目(代码)名称: ( 837 ) 数据结构与数据库系统

满分: 150

(考生注意: 答卷封面需填写自己的准考证编号, 答完后连同本试题一并交回!)

答题注意: 请在答题纸上按题目顺序答题, 在试题上答题无效!

### 第一部分 数据结构 (90 分)

#### 一. 选择题 (共 20 分, 10 小题, 每题 2 分)

1. 下列程序段的时间复杂度是 ( )。

```
for (i=0; i<n; i++)  
    for (j=1; j<m; j++)  
        A[i][j]=(i+1)*j;
```

A.  $O(n)$       B.  $O(m+n+1)$       C.  $O(m+n)$       D.  $O(m*n)$

2. 在长度为  $n$  的顺序表中删除第  $i$  个元素 ( $1 \leq i \leq n$ ) 时, 需移动的元素个数为 ( )。

A.  $i$       B.  $i+1$       C.  $n-i$       D.  $n-i+1$

3. 若要在  $O(1)$  的时间复杂度上, 通过两个单向循环链表的头尾相接实现合并, 则应对两个循环链表各设置一个指针, 分别指向 ( )。

A. 各自的头结点      B. 各自的尾结点  
C. 各自的第一个元素结点      D. 一个表的头结点, 另一个表的尾结点

4. 若进栈序列为  $a, b, c$ , 则通过入出栈操作可能得到的  $a, b, c$  的不同排列的个数为 ( )。

A. 4      B. 5      C. 6      D. 7

5. 已知循环队列的存储空间为数组  $data[21]$ , 且当前队列的头指针和尾指针的值分别为 8 和 3, 则该队列的当前长度为 ( )。

A. 5      B. 6      C. 16      D. 17

6. 已知二叉树的先序序列为  $abdecf$ , 中序序列为  $dbaefc$ , 则后序序列为 ( )。

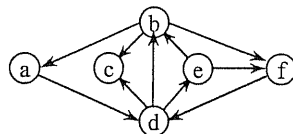
A.  $debfc a$       B.  $deba f c$       C.  $debc f a$       D.  $defb c a$

7. 若采用孩子兄弟链表作为树的存储结构, 则树的后序遍历应采用二叉树的【 】

A. 层次遍历算法      B. 前序遍历算法      C. 中序遍历算法      D. 后序遍历算法

8. 已知某有向图如右所示, 从顶点  $a$  出发, 不可能得到的深度优先遍历序列为 ( )。

A.  $adbefc$       B.  $adcefb$       C.  $adcbfe$       D.  $adefcb$



9. 设顺序存储的线性表共有 123 个元素, 按分块查找的要求等分成 3 块。若对索引表采用顺序查找来确定块, 并在确定的块中进行顺序查找, 则在查找概率相等的情况下, 分块查找成功时的平均查找长度为 ( )。

A. 21      B. 23      C. 41      D. 62

10. 在最好和最坏情况下的时间复杂度均为  $O(n \log n)$  且稳定的排序方法是 ( )。

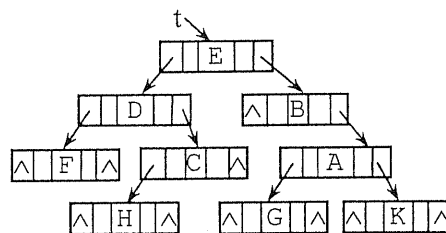
A. 快速排序      B. 堆排序      C. 归并排序      D. 基数排序

## 二. 解答题 (共 30 分, 4 小题, 每题 7 或 8 分)

11. (7 分) 已知某二叉树的二叉链表  $t$  如右图所示,

(1) 请写出该二叉树的后序遍历序列;

(2) 画出该二叉链表的后序线索链表。



12. (7 分) 已知有向图  $G$  定义如下:

$$G = (V, E)$$

$$V = \{a, b, c, d, e, f\}$$

$$E = \{ \langle a, b \rangle, \langle a, c \rangle, \langle a, d \rangle, \langle c, b \rangle, \langle c, e \rangle, \langle d, e \rangle, \langle f, d \rangle, \langle f, e \rangle \}$$

(1) 画出图  $G$  的逻辑结构;

(2) 写出  $G$  的 8 个拓扑序列。

13. (8 分) 设哈希函数为  $H(k) = k \text{ MOD } 11$ , 用二次探测法处理冲突。请画出依次插入元素 29, 15, 48, 47, 23, 41, 73, 37 后该哈希表的状态, 在各元素下面标出其冲突次数, 并求出查找成功的平均查找长度。

14. (8 分) 对关键字序列 (69, 88, 54, 23, 96, 11, 05, 47) 进行堆排序, 使之按关键字降序排列。请分别写出堆排序过程中得到的初始堆和第 1、2 趟的序列状态。

## 三. 算法填空题 (共 20 分, 4 小题, 每空 2 分)

15. (6 分) 算法  $f15$  的功能是, 对以带头结点的单链表作为存储结构的两个递增有序表  $La$  和  $Lb$  (表中不存在值相同的数据元素) 进行如下操作: 依次检查  $Lb$  表中的元素, 如果已在  $La$  中出现则删除之, 否则将元素结点插入到  $La$  中。请在空缺处填入合适的内容, 使其成为完整的算法。

```

void f15(LinkList La, LinkList Lb) {
    LinkList pre=La, q;
    LinkList pa=La->next, pb=Lb->next;
    free(Lb);
    while (pa && pb) {
        if (pa->data < pb->data)
            { pre=pa; pa=pa->next; }
        else if (pa->data > pb->data) {
            ① _____;
            pre=pb; pb=pb->next;
            ② _____;
        }
        else { q=pb; pb=pb->next; free(q); }
    }
    if (pb)
        ③ _____;
}
    
```

16. (4分) 如果二叉树  $T$  不含度为 1 的结点, 则称为正则二叉树。算法  $f16(T)$  判定二叉树  $T$  是否为正则二叉树, 如果是则返回 1, 否则返回 0。请在空缺处填入合适内容, 使其成为完整的算法。

```
int f16(BiTree T) {
    BiTree p; Queue Q;
    if ( ① ) return 1;
    InitQueue(Q); EnQueue(Q, T);
    do {
        DeQueue(Q, p);
        if (p->lchild || p->rchild)
            if ( ② ) {
                EnQueue(Q, p->lchild);
                EnQueue(Q, p->rchild);
            } else return 0;
    } while (!QueueEmpty(Q));
    return 1;
}
```

17. (6分) 已知带头结点的单链表  $L$  中的关键字为整数。算法  $f17$  为提高查找效率, 将它改建为采用拉链法处理冲突的散列表  $H$ 。设散列表  $H$  的长度为  $m$ , 散列函数为  $\text{Hash}(\text{key}) = \text{key} \% m$ 。链表的结点结构为:
- |     |      |
|-----|------|
| key | next |
|-----|------|

请在空缺处填入适当内容, 使其成为完整的算法。

```
void f17(LinkList L, LinkList H[], int m, int i) {
    // 由带头结点的单链表 L 生成散列表 H, 散列表生成之后原链表不再存在
    int i, j;
    LinkList p, q;
    for (i=0; i<m; i++)
        H[i] = ①;
    p = L->next;
    while(p) {
        q = p->next;
        j = p->key % m;
        ②;
        H[j] = p;
        ③;
    }
    free(L);
}
```

18. (4 分) 图的邻接表存储结构的类型定义如下:

```
typedef struct ArcNode {  
    int      adjvex;    // 该弧所指向的顶点的位置  
    ArcNode *nextArc;   // 指向下一条弧的指针  
} ArcNode;  
  
typedef struct {  
    VertexType data;    // 顶点信息  
    ArcNode *firstArc;  // 指向第一条依附该顶点的弧  
} VNode, AdjList[MAX_VERTEX_NUM]; // 定义顶点数组  
  
typedef struct {  
    AdjList vertices;  
    int      vexNum, arcNum; // 图的当前顶点数和弧数  
    int      kind;  
} ALGraph; // 邻接表类型
```

算法 f18 从有向图 G 中删除所有指向第 i 顶点的弧。请在空缺处填入合适内容, 使其成为完整的算法。

```
void f18(ALGraph &G, int i) {  
    int j;  
    ArcNode *p, *q;  
    for (j = 0; j < G.vexNum; j++) {  
        p = G.vertices[j].firstArc;  
  
        while ( _____ ① ) {  
            q = p;  
            p = p->nextArc;  
        }  
        if (p != NULL) {  
            if (p != G.vertices[j].firstArc)  
                q->nextArc = p->nextArc;  
            else _____ ② ;  
            free(p);  
            G.arcNum--;  
        }  
    }  
}
```

#### 四. 算法分析题 (共 20 分, 4 小题, 每题 5 分)

19. (5 分) L 是带头结点的双向循环链表的头指针, 阅读算法 f19, 并回答下列问题:

- (1) 当链表中结点个数分别为 1 和 4 (不包括头结点) 时, 请分别写出算法中 while 循环体的相应执行次数;
- (2) 简述算法 f19 的功能。

```
int f19(DLinkList L) {  
    DLLNode *p, *q;  
    int j=1;  
    p=L->next;  
    q=L->prior;  
    while(p!=q && p->prior!=q && j!=0)  
        if(p->data==q->data) {  
            p=p->next;  
            q=q->prior;  
        }  
    else j=0;  
    return j;  
}
```

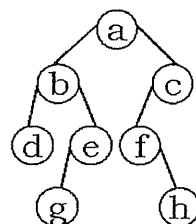
20. (5 分) 阅读算法 f20, 并回答下列问题:

- (1) 设队列 Q=(1, 3, 5, 4, 2), 写出执行算法 f20(Q) 后的队列 Q;
- (2) 简述算法 f20 的功能。

```
void f20(Queue Q)  
{ ElémType e;  
  if (!QueueEmpty(Q))  
  { e=DeQueue(Q);  
    f20(Q);  
    EnQueue(Q, e);  
  }  
}
```

21. (5 分) 阅读算法 f21, 并回答问题:

- (1) 已知如图所示的二叉树采用二叉链表存储结构, T 为指向根结点的指针。写出执行 f21(T) 的输出结果;
- (2) 简述算法 f21 的功能, 以及语句 “Push(S, NULL);” 的作用。



```

void f21(BinTree T){
    Stack S;
    if(T){
        InitStack(S);
        Push(S,NULL);
        while (T){
            printf("%c",T->data);
            if(T->rchild) Push(S,T->rchild);
            if(T->lchild) T=T->lchild;
            else T=Pop(S);
        }
    }
}

```

22. (5分) 图的邻接矩阵表示描述如下:

```

#define MaxNum 20      //图的最大顶点数
typedef struct {
    char vexs[MaxNum]; //字符类型的顶点表
    int edges[MaxNum][MaxNum]; //邻接矩阵
    int n,e; //图中当前的顶点数和边数
} MGraph; //邻接矩阵结构描述

```

阅读算法 f22, 并回答问题:

(1) 对于下列图 G 的邻接矩阵, 写出函数调用 f22(G,3) 的返回值;

$$\begin{bmatrix}
 0 & 1 & 1 & 1 & 1 \\
 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 \\
 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 0
 \end{bmatrix}$$

(2) 简述算法 f22 的功能。

```

int f22(MGraph G, int i) {
    int d=0,j;
    for (j=0; j<G.n; j++){
        if (G.edges[i][j]) d++;
        if (G.edges[j][i]) d++;
    }
    return d;
}

```

## 第二部分 数据库系统 (60 分)

### 一、选择题 (共 10 分, 5 小题, 每题 2 分)

- 1、下面对于数据(Data)、数据库(DB)、数据库系统(DBS)以及数据库管理系统(DBMS)等概念的描述中, 不正确的是 ( )。
- A. 数据库系统是人机系统, 数据库是数据库系统的一个组成部分。  
B. 数据是数据库管理的基本对象。  
C. 数据库管理系统由数据库和数据库系统构成。  
D. 微软公司的 SQL Server 软件产品是一个关系数据库管理系统。
- 2、描述数据库中全体数据的逻辑结构和特征是 ( )。
- A. 内模式      B. 模式      C. 外模式      D. 存储模式
- 3、关系数据库规范化是为了解决关系数据库中 ( ) 问题而引入的。
- A. 插入异常,删除异常和数据冗余      B. 提高查询速度  
C. 减少数据操作的复杂性      D. 保证数据的安全性和完整性
- 4、一般情况下, 一个关系中只有一个 ( )。
- A. 候选键      B. 主键      C. 超键      D. 外键
- 5、数据库中只存放视图的 ( )。
- A. 操作      B. 对应的数据      C. 定义      D. 限制

### 二、简答题 (共 20 分, 2 小题, 每题 10 分)

- 6、什么是数据库的完整性? 请给出关系数据库中三类完整性约束的基本概念(即约束规则), 并请说明下述 SQL 语句中分别定义了哪些完整性约束, 请根据其行号分别予以指出。

```
1 CREATE TABLE SC
2 (Sno CHAR(9),
3  Cno CHAR(4),
4  Grade SMALLINT CHECK(Grade>=0 AND Grade<=100),
5  PRIMARY KEY (Sno, Cno),
6  FOREIGN KEY(Sno) REFERENCE Student(Sno),
7  FOREIGN KEY(Cno) REFERENCE Course(Cno));
```

- 7、何谓数据库的三级模式和二级映像? 请说明二级映像的作用。

### 三、分析设计题 (共 30 分, 3 小题, 每题 10 分)

- 8、请为一个局部应用设计一个数据库, 它将保存每个部门及其职工的基本信息。已经通过数据抽象方式获得了两个实体: “职工”和“部门”, 一个职工只能属于某一个部门, 而一个部门有多个职工,

每个部门都有一名部门经理。其中，职工的基本信息包括：职工工号(ENO)、姓名(ENAME)、年龄(AGE)、工资(SALARY)和工作部门(DPT)；部门的基本信息包括：部门编号(DNO)、部门名称(DNAME)等。

画出上述实体间的 E-R 图，并根据 E-R 图，写出转换后的关系模型(其中1:1, 1:n关系均不转换为独立的关系模式)，并指出每个关系的主键和外键(如果存在的情况下)。

9、根据上述第 1 题中的基本关系表完成下述各题。

(1) 建立一个年龄大于 45 岁的职工视图 EMPV，属性为 (DNO, DNAME, ENO, ENAME, AGE, SALARY)。

(2) 写出以下查询的关系代数表达式和 SQL 语句：

检索每个部门经理的工资，要求显示其部门编号、部门名称、经理工号、经理姓名和经理工资。

10、已知关系 R 和关系 S 如下图所示，根据关系 R 和关系 S 的内容完成下述各题。

R

A	B
a	d
d	a
c	c
d	c

S

A	B
d	a
b	a
d	c

- (1) 求  $R_1 = R \cap S$ 。
- (2) 求  $R_2 = R \times S$ 。
- (3) 求  $R_3 = R - S$ 。