

# 广东工业大学

## 2013 年攻读硕士学位研究生入学考试试题

考试科目（代码）名称：(829) 数据结构

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！）

### 一. 选择题（共 30 分，15 小题，每题 2 分）

1. 下面程序段的时间复杂度是（ ）。

```
x = 1;
for(i = 1; i < n; i++)
    for(j = 1; j < i; j++)
        x++;
```

- A.  $O(1)$                       B.  $O(n)$                       C.  $O(n^2)$                       D.  $O(\log_2 n)$

2. 对于长度为  $n$  的顺序表，假定删除表中任一元素的概率相同，则删除一个元素平均需要移动元素的个数是（ ）。

- A.  $n$                       B.  $n/2$                       C.  $(n-1)/2$                       D.  $(n+1)/2$

3. 顺序表具有的特点是（ ）。

- A. 不必事先估计存储空间                      B. 可以随机访问任一结点  
C. 存储空间不一定要连续                      D. 插入和删除不需要移动元素

4. 如果一个栈的输入序列为 12345，其不可能的输出序列是（ ）。

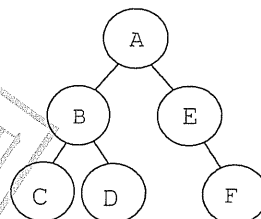
- A. 14352                      B. 23415                      C. 31245                      D. 43521

5. 假设用数组  $A[8]$  存储循环队列的元素，其头、尾指针  $front$  和  $rear$  的当前值分别为 4 和 0。当从队列中出队列两个元素，再入队列一个元素后， $front$  和  $rear$  的值分别为（ ）。

- A. 3 和 6                      B. 6 和 3                      C. 1 和 6                      D. 6 和 1

6. 若对如图所示的二叉树进行中序线索化，则结点 D 的前驱和后继线索分别指向（ ）。

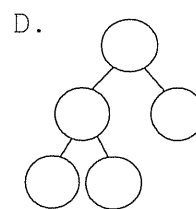
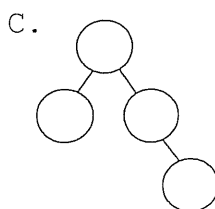
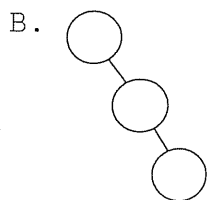
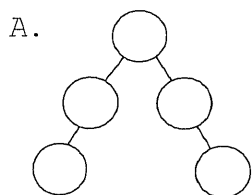
- A. 结点 B 和结点 A                      B. 结点 C 和结点 B  
C. 结点 C 和结点 E                      D. 结点 B 和结点 E



7. 在一棵二叉树中，度为 2 的结点有 15 个，度为 1 的结点有 2 个，则度为 0 的结点数为（ ）。

- A. 13                      B. 15                      C. 16                      D. 17

8. 下列二叉树中, 不满足二叉平衡树定义的是 ( )。

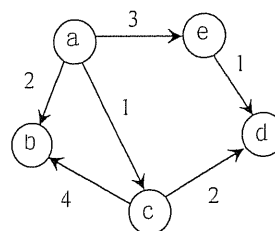


9. 下列关于图的叙述中, 正确的是 ( )。

- A. 用邻接矩阵存储图, 所占存储空间大小只与图中结点数有关, 而与边数无关
- B. 用邻接矩阵存储图, 所占存储空间大小只与图中边数有关, 而与结点数无关
- C. 用邻接表存储图, 所占存储空间大小只与图中结点数有关, 而与边数无关
- D. 用邻接表存储图, 所占存储空间大小只与图中边数有关, 而与结点数无关

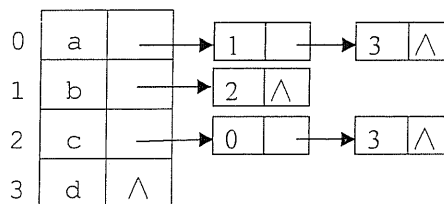
10. 已知带权图 G 如图所示, 若采用迪杰斯特拉算法求源点 a 到其它顶点的最短路径, 则得到的第一条最短路径的目标顶点是 ( )。

- A. 顶点 b
- B. 顶点 c
- C. 顶点 d
- D. 顶点 e



11. 已知有向图 G 的邻接表如图所示, 基于该邻接表, 可求得从顶点 a 出发的深度优先遍历序列是 ( )。

- A. abdc
- B. abcd
- C. acdb
- D. acbd



12. 对有序表  $(a_1, a_2, a_3, a_4, a_5, a_6)$  进行折半查找, 查找元素  $a_2$  共需进行比较的次数为 ( )。

- A. 1
- B. 2
- C. 3
- D. 4

13. 以下有关 m 阶 B-树的叙述中, 错误的是 ( )。

- A. 根结点最多有 m 棵子树
- B. 所有叶子结点都在同一层上
- C. 各结点内关键字均升序或降序排列
- D. 叶子结点之间通过指针链接

14. 若对关键字序列  $(42, 70, 53, 28, 40, 84)$  以第一个关键字为枢轴进行一趟快速排序, 则得到的结果为 ( )。

- A.  $(28, 40, 42, 53, 70, 84)$
- B.  $(40, 28, 42, 70, 53, 84)$
- C.  $(40, 28, 42, 53, 70, 84)$
- D.  $(40, 28, 42, 84, 53, 70)$

15. 要在  $O(n \log n)$  时间内对数组进行稳定的排序, 可用的排序方法是 ( )。

- A. 快速排序
- B. 归并排序
- C. 冒泡排序
- D. 堆排序

## 二. 填空题 (共 20 分, 10 小题, 每题 2 分)

1. 根据数据元素之间关系的不同特性, 数据结构可分为四种基本类型: 集合结构、线性结构、和\_\_\_\_\_。
2. 线性表的表长是指\_\_\_\_\_。
3. 已知无头结点的单链表的头指针为  $L$ , 则判断该单链表为空的条件是\_\_\_\_\_。
4. 在队列中, 允许插入的一端称为\_\_\_\_\_, 允许删除的一端称为\_\_\_\_\_。
5. 已知某完全二叉树的第 4 层 (设根为第 1 层) 只有 1 个结点, 则该树共有\_\_\_\_\_个叶子结点。
6. 设将一棵完全二叉树顺序存储在一维数组  $A[1..n]$  中, 若  $A[i]$  的右孩子结点存在, 则该结点应存储在  $A[_____]$  位置上。
7. 具有  $n$  个顶点的有向图最多有\_\_\_\_\_条边。
8. 若在无向图  $G$  的存储结构中, 顶点表  $V = (a, b, c, d)$ , 邻接矩阵  $A$  如图所示, 则顶点  $a$  的度是\_\_\_\_\_。
$$A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$
9. 若对二叉排序树进行\_\_\_\_\_遍历, 则可得到该二叉排序树所有结点的有序序列。
10. 在排序中, 从未排序序列中依次取出元素与已排序序列 (初始序列为空) 中的元素进行比较, 将其放入已排序列的适当位置, 则称该方法为\_\_\_\_\_。

## 三. 解答题 (共 42 分, 6 小题, 每题 7 分)

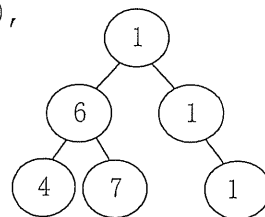
1. (7 分) 已知二叉树的层次遍历序列为 ABCDE, 中序遍历序列为 BDACE, 请画出该二叉树。

2. (7 分) 已知如图所示的二叉排序树, 依次删除关键字 4, 6, 10, 试画出该树删除过程中的以下状态:

(1) (2 分) 删除 4 之后;

(2) (2.5 分) 删除 6 之后;

(3) (2.5 分) 删除 10 之后。



3. (7 分) 已知在无向图  $G$  的存储结构中, 顶点表  $V = (a, b, c, d, e)$ , 由于其邻接矩阵为对称矩阵, 可以只将邻接矩阵的下三角元素 (包含主对角线元素) 按行序为主序的顺序依次保存在如下的连续存储区域中:

0	1	0	1	0	0	1	1	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(1) (4 分) 写出图  $G$  的邻接矩阵;

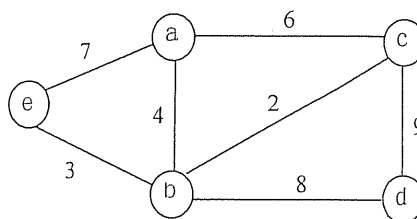
(2) (3 分) 画出无向图  $G$ 。

4. (7分) 已知如图所示的无向图 G, 试按照克鲁斯卡尔算法求图 G 的最小生成树:

(1) (3分) 画出该最小生成树;

(2) (4分) 写出依次选取的各条边。

(注: 每条边的书写格式为 “a—4—b”)



5. (7分) 设哈希函数为  $H(k) = (k^2 + 1) \% 7$ , 用链地址法处理冲突。

(1) (5分) 画出依次插入元素 4, 3, 8, 7, 6 后, 该哈希表的状态;

0	1	2	3	4	5	6

(2) (2分) 求查找成功时的平均查找长度。

6. (7分) 对序列 (50, 80, 63, 96, 22, 31, 55, 11) 执行升序的希尔排序算法, 增量序列为 (5, 3, 1), 写出排序中第二趟的结果。

第一趟: (31, 55, 11, 96, 22, 50, 80, 63)

第二趟: ( )

第三趟: (11, 22, 31, 50, 55, 63, 80, 96)

#### 四. 算法阅读题 (共 24 分, 3 小题, 每题 8 分)

1. (8分) 设 L 为带头结点的单链表, 阅读算法 f1, 回答下列问题:

(1) (4分) 若  $L = (2, 5, 4, 6, 3)$ , 请写出执行算法 f1(L) 后的返回值;

(2) (4分) 简述算法 f1 的功能。

```
int f1(LinkList L) {
    for(i = 0, p = L->next; p ; p=p->next)
        i++;
    return i;
}
```

2. (8分) 阅读算法 f2, 回答下列问题:

(1) (4分) 设栈  $S = (3, 6, 4, 5, 8, 7)$ , 栈顶元素为 7, 请写出执行算法 f2(S, 5) 后的 S;

(2) (4分) 简述算法 f2 的功能。

```
void f2(Stack &S, ElemType e) {
    InitStack(T);
    while(!StackEmpty(S)) {
        Pop(S, d);
        if(d > e) Push(T, d);
    }
    while(!StackEmpty(T)) {
        Pop(T, d);
        Push(S, d);
    }
}
```

3. (8分) 设顺序表 L 中的数据元素递增有序, 阅读算法 f3, 回答下列问题:

(1) (4分) 设  $L = (3, 5, 9, 12, 16)$ , 请分别写出执行算法  $f3(L, 12)$  和  $f3(L, 15)$  后的返回值;

(2) (4分) 简述算法 f3 的功能。

```
int f3(SqList L, ElemType e) {  
    l = 0; h = L.length-1;  
    while(l <= h) {  
        m = (l+h)/2;  
        if(e == L.elem[m])  
            return m;  
        else  
            if(e < L.elem[m])  
                h = m-1;  
            else  
                l = m+1;  
    }  
    return -1;  
}
```

## 五. 算法填空题 (共 24 分, 3 小题, 每题 8 分)

1. (8分) 循环链表的类型定义如下:

```
typedef struct LNode {  
    ElemType data;  
    struct LNode *next;  
} LNode, *LinkList;
```

算法 f4 实现队列中的出队列操作, 其中队列用带头结点的循环链表表示, 并且队列只设一个指针 rear 指向队尾元素结点 (注意不设头指针)。请在空缺处填入合适内容, 使其成为完整的算法。

```
Status f4(LinkList &rear, ElemType &e) {  
    if(①) return ERROR;  
    p = ②;  
    e = p->data;  
    rear->next->next = ③;  
    if(p == rear)  
        ④;  
    free(p);  
    return OK;  
}
```

2. (8 分) 二叉树的二叉链表存储结构的类型定义如下:

```
typedef struct BiTNode {  
    char data;  
    struct BiTNode *lchild, *rchild;  
}BiTNode, *BiTree;
```

算法 f5 求二叉树中非叶子结点总数, 其中二叉树采用二叉链表做存储结构。请在空缺处填入合适内容, 使其成为完整的算法。

```
int f5(BiTree T) {  
    if( ① )  
        return 0;  
    if(T->lchild==NULL && ② )  
        return 0;  
    return f5(T->lchild) + ③ + ④ ;  
}
```

3. (8 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; // 定义弧的结点  
  
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 f6 在邻接表存储结构上实现图的深度优先遍历。请在空缺处填入合适内容, 使其成为完整的算法。

```
void f6(ALGraph G, int i) {  
    visit(G.vertices[i].data);  
    visited[i]=TRUE;  
    for(p = ① ; p ; p = ② )  
        if(!visited[③ ] )  
            f6(G, ④ );  
}
```

## 六. 算法设计题（共 10 分，1 题）

设线性表  $L$  中的数据元素递增有序，采用带头结点的单链表做存储结构，其类型定义如下：

```
typedef struct LNode {  
    ElemType data;  
    struct LNode *next;  
}LNode, *LinkList;
```

试写算法，将元素  $x$  插入到线性表  $L$  中，并保持该表的有序性。

