

2016 年 10 月高等教育自学考试全国统一命题考试

数据结构导论 试卷

(课程代码 02142)

本试卷共 4 页，满分 100 分，考试时间 150 分钟。

考生答题注意事项：

1. 本卷所有试题必须在答题卡上作答。答在试卷上无效，试卷空白处和背面均可作草稿纸。
2. 第一部分为选择题。必须对应试卷上的题号使用 2B 铅笔将“答题卡”的相应代码涂黑。
3. 第二部分为非选择题。必须注明大、小题号，使用 0.5 毫米黑色字迹签字笔作答。
4. 合理安排答题空间。超出答题区域无效。

第一部分 选择题(共 30 分)

一、单项选择题(本大题共 10 小题，每小题 2 分，共 30 分)

在每小题列出的四个备选项中只有一个是符合题目要求的，请将其选出并将“答题卡”的相应代码涂黑。错涂、多涂或未涂均无分。

1. 已知问题规模为 n ，则下列程序片段的时间复杂度是

```
i=1;j=0;
while(i+j<=n){if(i>j)j++;else i++;}
```

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

2. 若用计算机来模拟银行客户排队等待办理业务的情形，则所应该采用的数据结构是

A. 栈 B. 队列 C. 树 D. 图

3. 若线性表采用链式存储结构，则适用的查找方法为

A. 随机查找 B. 散列查找 C. 二分查找 D. 顺序查找

4. 已知指针 P 和 q 分别指向某单链表中第一个结点和最后一个结点，假设指针 s 指向另一个单链表中某个结点，则在 S 所指结点之后插入上述单链表应执行的语句为

A. $q \rightarrow next = s \rightarrow next; s \rightarrow next = P;$ B. $s \rightarrow next = P; q \rightarrow next = s \rightarrow next;$
C. $p \rightarrow next = s \rightarrow next; s \rightarrow next = q;$ D. $s \rightarrow next = q; p \rightarrow next = s \rightarrow next;$

5. 栈的运算特点是先进后出，元素 a 、 b 、 c 、 d 依次入栈，则不能得到的出栈序列是

A. $abed$ B. $dcba$ C. $cabd$ D. $bcda$

6. 在实现队列的链表结构中，其时间复杂度最优的是

A. 仅设置头指针的单循环链表 B. 仅设置尾指针的单循环链表
C. 仅设置头指针的双向链表 D. 仅设置尾指针的双向链表

7. 任意一棵二叉树的前序和后序遍历的结果序列中，各叶子结点之间的相对次序关系是

A. 不一定相同 B. 都相同 C. 都不相同 D. 互为逆序

8. 若某棵树的存储结构采用双亲表示法，如题 8 图所示，则该树的高度是

数组下标	data	parent
1	R	0
2	A	1
3	B	1
4	C	1
5	D	2
6	E	2
7	F	4
8	G	7
9	H	7
10	K	7

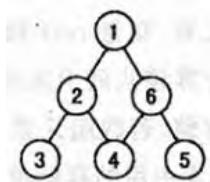
题 8 图

- A. 2 B. 3 C. 4 D. 5

9. 无向图的邻接矩阵一定是

- A. 对称矩阵 B. 对角矩阵 C. 稀疏矩阵 D. 三角矩阵

10. 根据连通图的深度优先搜索的基本思想, 如题 10 图所示的连通图的一个深度优先搜索的结果序列是



题 10 图

- A. 123456 B. 123465 C. 126345 D. 162543

11. 用顺序查找方法对含有 n 个数据元素的顺序表按从后向前查找次序进行查找, 现假设查找

其中每个数据元素的概率不相等, 那么

- A. 该顺序表按查找概率由低到高的顺序来存储数据元素, 其 ASL 最小
 B. 该顺序表按查找概率由高到低的顺序来存储数据元素, 其 ASL 最小
 C. ASL 的大小与数据元素在该顺序表中的位置次序无关
 D. ASL 的大小与查找每个数据元素的概率无关

12. 已知散列表的存储空间为 $T[0, \dots, 16]$, 散列函数为 $H(k) = k \bmod 17$, 用二次探测法解决冲突。散列表中已插入下列关键字: $T[5] \leftarrow 39$ 、 $T[6] \leftarrow 57$ 和 $T[7] \leftarrow 7$, 则下一个关键字值 23 在该散列表中插入的位置是

- A. $T[23]$ B. $T[4]$ C. $T[8]$ D. $T[10]$

13. 对关键字序列 {eSC, tab, ah, con, brk, del} 进行排序时, 若关键字序列的变化情况如下: ①esc, tab, ah, con, brk, del ②ah, tab, eSC, con, brk, del ③alt, brk, esc, con, tab, del ④alt, brk, con, esc, tab, del ⑤ah, brk, con, del, tab, esc ⑥ah, brk, con, del, esc, tab。则所用的排序方法是

- A. 直接插入排序 B. 直接选择排序 C. 堆排序 D. 冒泡排序

14. 满足最小堆定义的是

- A. {21, 25, 55, 23, 51, 63} B. {21, 51, 55, 63, 25, 23}
 C. {21, 63, 55, 25, 51, 23} D. {21, 51, 23, 63, 55, 25}

15. 设有两个长度分别为 m 、 n 的降序有序序列 $\{a_1, a_2, \dots, a_m\}$ 、 $\{b_1, b_2, \dots, b_n\}$, 采用二路归并方法将它们合并成长度为 $m+n$ 的降序有序序列, 则归并过程中元素比较次数最少的

条件一定是

A. $a_1 > b_1$

B. $a_m > b_n$

C. $a_1 < b_n$

D. $a_m < b_1$

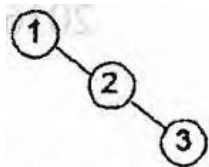
第二部分非选择题(共 70 分)

二、填空题(本大题共 13 小题, 每小题 2 分, 共 26 分)

16. 从宏观上看, 数据、数据元素和_____反映了数据组织的三个层次。
17. 在表长为 n 的顺序表中插入或删除一个元素, 则需移动元素的具体个数与表长和_____有关。
18. 非空的单循环链表的头指针为 $head$, 尾指针为 $rear$, 则 $rear \rightarrow next =$ _____。
19. 设以数组 $Q[m]$ 存放循环队列的元素, 变量 $rear$ 和 $queuelen$ 分别表示循环队列中队尾元素的下标位置和元素的个数。则计算该队列中队头元素下标位置的公式是_____。
20. 二维数组 $A[8][9]$ 按行优先顺序存储, 若数组元素 $A[2][3]$ 的存储地址为 1087, $A[4][7]$ 的存储地址为 1153, 则每个数组元素占用的存储单元的个数是_____。
21. 设一个完全二叉树共含有 196 个结点, 则该完全二叉树中含有叶结点的个数是_____。
22. 假设高度为 h 二叉树中只有度为 2 和度为 0 这两种类型的结点, 则该类二叉树中结点个数至多为 $2^h - 1$ 、至少为_____。
23. 若以数据集 $\{34, 5, 12, 23, 8, 18\}$ 为叶结点的权值构造一棵哈夫曼(Huffman)树, 那么该 Huffman 树的带权路径长度 WPL_____。
24. 设有散列函数 $H(k)$ 和键值 $k_1, k_2 (k_1 \neq k_2)$, 若 $H(k_1) = H(k_2)$, 则这种现象称为“冲突”, 且称键值 k_1 和 k_2 互为_____。
25. 一个图的最小生成树是满足一定条件的生成树, 即一个图的最小生成树是指该图的所有生成树中_____的生成树。
26. 对长度为 n 的有序顺序表进行二分查找, 则查找表中的任意一个元素时, 无论查找成功与失败, 最多与表中_____个元素进行比较。
27. 排序方法中, 从未排序序列中依次取出元素与已排序序列(初始时空)中的元素按序进行比较, 将其插入已排序序列的正确位置上的方法称为_____。
28. 一般情况下, 时间复杂度是 $O(n \log_2 n)$ 且其空间复杂度最优的排序方法是_____。

三、应用题(本大题共 5 小题, 每小题 6 分, 共 30 分)

29. 借助于队列能够将含有 n 个数据元素的栈逆置, 比如栈 S 中的元素为 $\{a, b, c\}$ 逆置后变成 $\{c, b, a\}$ 。试简述你的解决方案。
30. 为便于表示二叉树的某些基本运算, 则深度为 k 的二叉树的顺序存储结构中的数组的大小为多少? 画出如题 30 图所示的二叉树的顺序存储结构示意图, 并说明对一般形态的二叉树不太适合使用顺序存储结构来表示的原因。



题 30 图

31. 先序遍历、中序遍历一个森林分别等同于先序、中序遍历该森林所对应的二叉树。现已知一个森林的先序序列和中序序列分别为 ABCDEFGJH 和 BDCAIFJGHE, 试画出该森林。
32. 设有一组关键字值序列 $\{e, b, d, f, a, g, C\}$ 现要求: (1) 根据二叉排序树的创建方法构造出相应的二叉排序树(关键字值的大小按字母表顺序计); (2) 计算等概率情况下在该二

叉排序树上查找成功的平均查找长度 ASL。

33. 若采用二路归并排序方法对关键字序列 {25, 9, 78, 6, 65, 15, 58, 18, 45, 20} 进行升序排序, 写出其每趟排序结束后的关键字序列。

四、算法设计题(本大题共 2 小题, 每小题 7 分, 共 14 分)

34. 某电商有关手机的库存信息, 按其价格从低到高存储在一个带有头结点的单循环链表中, 链表中的结点由品牌型号 (nametype)、价格 (price)、数量 (quantity) 和指针 (next) 四个域组成。现新到 in 台、价格为 c、品牌型号为 x 的新款手机需入库, 写出相应的存储结构和实现该要求的算法。

35. 写出向存储结构为邻接矩阵的无向图 G 中插入一条边 (x, y) 的算法。算法的头函数为: void AddEdgetoGraph(Graph*G, VertexType X, VertexType y>, 无向图 G 的存储结构为:

```
#define MaxVertex num
typedef char VertexType;
typedef int EdgeType;
typedef struct graph {
    int n, e; // 图的实际顶点数和边数
    EdgeType edge [MaxVertex][MaxVertex]; // 邻接矩阵
    VertexType vertex[MaxVertex]; // 顶点表
} Graph;
```

一、单项选择题(本大题共15小题,每小题2分,共30分)

- | | | | | |
|-------|-------|-------|-------|-------|
| 1. C | 2. B | 3. D | 4. A | 5. C |
| 6. B | 7. B | 8. C | 9. A | 10. B |
| 11. A | 12. D | 13. B | 14. D | 15. C |

二、填空题(本大题共13小题,每小题2分,共26分)

- | | |
|------------------------------------|--|
| 16. 数据项 | 17. 该元素所处的位置 |
| 18. head | 19. $(\text{rear} - \text{queuelen} + m) \% m$ |
| 20. 3 | 21. 98 |
| 22. $2h-1$ | 23. 238 |
| 24. 同义词 | 25. 权值之和最小 |
| 26. $\lfloor \log_2 n \rfloor + 1$ | 27. 直接插入排序 |
| 28. 堆排序 | |

三、应用题(本大题共5小题,每小题6分,共30分)

29. 先将栈中元素依次出栈并入队列,(3分)然后使该队列元素依次出队列并进入栈。(3分)

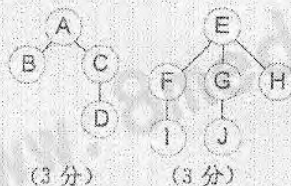
30. 数组的大小为 $2^k - 1$;(2分)顺序存储结构示意图:

1	0	2	0	0	0	3
---	---	---	---	---	---	---

;(2分)原

因:会造成存储空间的浪费现象。(2分)

31. 先根据给定的两个序列构造出相应的二叉树,然后再将其转成森林:

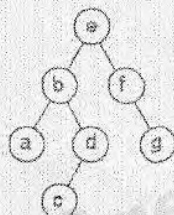


(3分)

(3分)

答 31 图

32. (1)构造出的二叉排序树如答 32 图。



答 32 图

(3分)

(2) $ASL = (1 \times 1 + 2 \times 2 + 3 \times 3 + 1 \times 4) / 7 = 18/7$ 。

(3分)

43. 初始态:[25][9][78][6][65][15][58][18][45][20]

第一趟:[9 25][6 78][15 65][18 58][20 45]

(2分)

第二趟:[6 9 25 78][15 18 58 65][20 45]

(1分)

第三趟:[6 9 15 18 25 58 65 78][20 45]

(1分)

第四趟:[6 9 15 18 20 25 45 58 65 78]

(2分)

四、算法设计题(本大题共2小题,每小题7分,共14分)

34. 存储结构为:

```
typedef struct node {  
    char * nametype; float price; int quantity;  
    struct node * next; } Node, * LinkedList;
```

(2分)

实现算法为:

```
void InsertData(LinkedList head, char * x, int m, float c)
```

```
{ new=(Node *)malloc(sizeof(Node));
```

```
new->nametype=x; new->price=c; new->quantity=m; new->next=NULL;
```

(1分)

```
q=head; p=head->next; //假设指针q指向p所指结点的前驱
```

```
while (p!=head && p->price<c) {q=q->next; p=p->next;}
```

(2分)

```
new->next=p; q->next=new; return;}
```

(2分)

35. void AddEdgetoGraph(Graph * G, VertexType x, VertexType y)

```
{ i=-1; j=-1;
```

```
for (k=0; k<G->n; k++) //查找x,y的编号
```

(3分)

```
{ if(G->vertex[k]==x) i=k;
```

```
if (G->vertex[k]==y) j=k;}
```

```
if (i==-1 || j==-1) Error("结点不存在");
```

(1分)

```
else { //插入边(x,y)
```

(3分)

```
G->edge[i][j]=1; G->edge[j][i]=1; G->e++;}
```

```
}
```