

广东工业大学

2015 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）

- 下列数据结构中属于线性结构的是（ ）。
A. 二叉树 B. 栈 C. 查找表 D. 图
- 已知带头结点的单链表的头指针为 L，删除第一个元素结点的语句是（ ）。
A. $L=L->next;$ B. $L->next=L->next->next;$
C. $L=L->next->next;$ D. $L->next=L;$
- 如果仅在线性表中删除第一个元素和在最后一个元素之后插入元素，宜采用的存储结构是（ ）。
A. 双向链表 B. 仅带头指针的单循环链表
C. 单链表 D. 仅带尾指针的单循环链表
- 如果一个栈的输入序列为 1234，则在可能的输出序列中，以 3 打头的序列个数是（ ）。
A. 2 B. 3 C. 4 D. 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 5 和 1。当从队列中出队一个元素和入队两个元素后，front 和 rear 的值分别为（ ）。
A. 2 和 7 B. 7 和 2 C. 6 和 3 D. 3 和 6
- 若某二叉树的先序和中序遍历序列相同，则该二叉树中（ ）。
A. 所有结点均无左孩子 B. 所有结点均无右孩子
C. 只包含根结点 D. 所有结点只有一个孩子
- 假设森林 F 对应的二叉树为 B，其包含 m 个结点，且根结点的右子树包含 n 个结点，则 F 中第一棵树的结点个数为（ ）。
A. $m-n$ B. $m-n-1$ C. $n+1$ D. n
- 下列关键字序列中，不能构成二叉排序树中一条查找路径的是（ ）。
A. 1, 4, 2, 3 B. 1, 2, 4, 3
C. 4, 3, 1, 2 D. 4, 2, 1, 3

9. 在某有向图的邻接矩阵 A 中, 图的第 i 个顶点的出度为 A 中 ()。

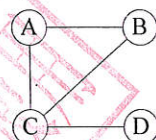
- A. 第 i 行非 ∞ 元素个数
B. 第 i 列非 ∞ 元素个数
C. 第 i 行非 ∞ 且非 0 元素个数
D. 第 i 列非 ∞ 且非 0 元素个数

10. 下列关于最小生成树的表述中, 正确的是 ()。

- A. 最小生成树的代价唯一
B. 权值最小的边一定会出现在所有的最小生成树中
C. 用普里姆算法从不同顶点开始得到的最小生成树一定不相同
D. 普里姆算法和克鲁斯卡尔算法得到的最小生成树总相同

11. 已知如图所示的无向图 G , 若从顶点 A 出发进行广度优先遍历, 则可能得到的遍历序列是 ()。

- A. ABDC
B. ADBC
C. ACBD
D. ACDB

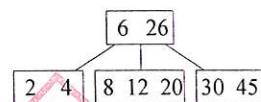


12. 在用线性探测法处理冲突的哈希表中查找某一关键字, 探测到的地址序列中的关键字值 ()。

- A. 都是同义词
B. 都不是同义词
C. 都相同
D. 不一定是同义词

13. 如图所示的是一棵 ()。

- A. 3 阶 B 树
B. 4 阶 B 树
C. 3 阶 B+ 树
D. 4 阶 B+ 树



14. 若关键字序列 (13, 6, 17, 32, 21) 是采用下列排序方法之一得到的第一趟排序后的结果, 则该排序算法是 ()。

- A. 冒泡排序
B. 简单选择排序
C. 快速排序
D. 2 路归并排序

15. 不需要进行关键字比较的排序方法是 ()。

- A. 快速排序
B. 堆排序
C. 归并排序
D. 基数排序

二. 填空题 (共 20 分, 10 小题, 每题 2 分)

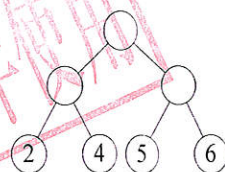
1. 在数据存储结构中, 除了要表示数据元素的值外, 还要表示_____。

2. 在长度为 n 的顺序表的表尾添加元素的算法的时间复杂度为_____。

3. 双向链表的主要优点是_____。

4. 队列的操作是按_____的原则进行的。

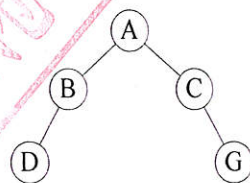
5. 如图所示的哈夫曼树的其带权路径长度 WPL 为_____。



6. 若高度为 h 的二叉树中只有度为 0 和 2 的结点, 则该树至多包含 _____ 个结点。
7. 若对包含 n 个顶点的无向图 G 进行深度优先遍历, 所得的遍历序列唯一, 则可断定 G 的边数为 _____。
8. 已知有向图 $G=(V, E)$, 其中 $V=\{a, b, c, d\}$, $E=\{<a, b>, <a, c>, <b, c>, <c, d>\}$, 图 G 的一个拓扑序列是 _____。
9. 在构造哈希表时, 不仅要设定一个好的哈希函数, 还要设定一种 _____。
10. 快速排序在最坏情况下的时间复杂度是 _____。

三. 解答题 (共 42 分, 6 小题, 每题 7 分)

1. (7 分) 对如图所示的二叉树 T 进行中序线索化。



2. (7 分) 从空树开始构造一棵二叉平衡树, 依次插入关键字为

12, 2, 6, 14, 17

请分别画出该树生成过程中的以下状态:

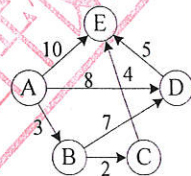
- (1) (2 分) 插入 12 和 2 之后;
- (2) (2 分) 插入 6 之后;
- (3) (3 分) 插入 14 和 17 之后。

3. (7 分) 已知有向图 G 的顶点集 $V=\{a, b, c, d\}$, 邻接矩阵 A 如图所示, 请:

- (1) (3 分) 画出有向图 G ;
- (2) (4 分) 画出图 G 的邻接表。

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

4. (7 分) 已知有向图 G 如图所示, 试按照迪杰斯特拉算法, 依次写出源点 A 到其它顶点的最短路径和最短路径长度 (顺序不能颠倒)。



5. (7 分) 已知关键字有序序列 (8, 17, 26, 32, 45, 60), 试画出对其进行折半查找的判定树, 并求出等概率情况下查找成功时的平均查找长度。

6. (7 分) 已知关键字序列 (31, 63, 22, 96, 50, 55), 试判断该序列是否是大顶堆, 若是请说明理由, 若不是请调整为大顶堆。

四. 算法阅读题 (共 24 分, 3 小题, 每题 8 分)

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

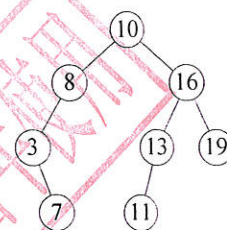
- (1) (4 分) 若 $L = (2, 8, 5, 1, 7, 6)$, 请写出执行算法 f1(L, 7) 后的返回值;
 (2) (4 分) 简述算法 f1 的功能。

```
int f1(LinkList L, ElemType e) {
    p = L->next; i = 1;
    while(p!=NULL && p->data!=e) { p = p->next; i++; }
    if(p!=NULL) return i;
    else return 0;
}
```

2. (8 分) 已知二叉排序树 T 采用二叉链表存储结构, 阅读算法 f2, 回答下列问题:

- (1) (4 分) 若 T 如图所示, 且 $k=0$, 请写出执行算法 f2(T, 15, k) 后的 k;
 (2) (4 分) 简述算法 f2 的功能。

```
void f2(BiTree T, ElemType e, int &k) {
    if(T!=NULL) {
        f2(T->lchild, e, k);
        if(T->data<=e) { k++; f2(T->rchild, e, k); }
    }
}
```



3. (8 分) 阅读算法 f3, 回答下列问题:

- (1) (4 分) 若顺序表 $L = (2, 5, 4, 6, 3)$, 请写出执行算法 f3(L) 后的 L;
 (2) (4 分) 简述算法 f3 的功能。

```
void f3(SqList &L) {
    for(i = 1; i<L.length; i++) {
        e = L.elem[i];
        l = 0; h = i-1;
        while(l<=h) {
            m = (l+h)/2;
            if(e<L.elem[m]) h = m-1;
            else l = m+1;
        }
        for(j = i-1; j>=h+1; j--) L.elem[j+1] = L.elem[j];
        L.elem[h+1] = e;
    }
}
```

五. 算法填空题 (共 24 分, 3 小题, 每题 8 分)

1. (8分) 设队列Q采用无头结点的链队列存储结构, 其类型定义如下:

```
typedef struct QNode {
    ElemType data;
    struct QNode *next;
} QNode, *QueuePtr;

typedef struct {
    QueuePtr front;
    QueuePtr rear;
} LinkQueue;
```

算法f4实现队列Q的入队操作。请在空缺处填入合适内容, 使其成为完整的算法。

```
Status f4(LinkQueue &Q, ElemType e) {
    p = (QueuePtr)malloc(sizeof(QNode));
    if(NULL==p) return OVERFLOW;
    p->data = e; ① = NULL;
    if(NULL==Q.front)
        ② = p;
    else
        ③ = p;
    ④;
    return OK;
}
```

2. (8分) 设二叉树T采用二叉链表存储结构, 其类型定义如下:

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

算法f5求二叉树T中叶子结点个数。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(NULL==T->lchild && ②)
        ③;
    return f5(T->rchild)+④;
}
```


3. (8 分) 设图 G 采用数组表示法存储结构, 其类型定义如下:

```
#define MaxNum 5
typedef struct {
    VexType vexs[MaxNum]; // 顶点表
    int arcs[MaxNum][MaxNum]; // 邻接矩阵
    int n,e; // 结点数和边数
} MGraph;
```

算法 f6 遍历图 G 中顶点 v 的所有邻接顶点。请在空缺处填入合适内容, 使其成为完整的算法。

```
void f6(MGraph G, VexType v, Status(*visit)(VexType)) {
    for(i = 0; i < G.n && ① != v; i++);
    if(②) return;
    for(j = 0; j < G.n; j++)
        if(③ != 0) visit(④);
}
```

六. 算法设计题 (共 10 分, 1 题)

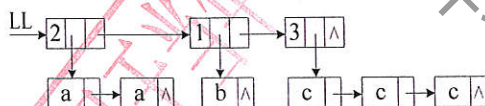
设有结点类型为 LNode 的可含有重复元素的非递减有序链表 L。请编写一个时间复杂度为 $O(n)$ 的算法, 将 L 分拆并重构为满足以下条件的有序链表 LL:

(1) 新建链表 LL 的结点类型为 SLNode, 含有 3 个域, 其中, 指针域 same 指向由 L 中值相同的结点组成的子链表, n 为子链表长度, 指针域 snext 指向 LL 中的下一个结点。

(2) LL 按 same 所指向的子链表的元素值非递减有序, 例如, 设有 L:



重构的结果 LL 为:



L 和 LL 链表及其结点的类型定义为

```
typedef struct LNode {
    ElemType data;
    struct LNode *next;
} LNode, *LinkList; // L 链表的结点和指针类型

typedef struct SLNode {
    int n;
    struct LNode *same;
    struct SLNode *snext;
} SLNode, SlinkList; // LL 链表的结点和指针类型
```