

广东工业大学

2019 年硕士学位研究生招生考试试题

考试科目（代码）名称：(829)数据结构

满分 150 分

（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. 与数据元素本身的形式、相对位置和个数无关的是（）。
A. 数据存储结构 B. 数据逻辑结构
C. 算法 D. 操作
2. 将长度为 n 和 m 的有序表合并为一个有序表，其最优算法的时间复杂度为（）。
A. $O(mn)$ B. $O(n)$ C. $O(m)$ D. $O(n+m)$
3. 若线性表最常用的操作是存取第 i 个元素及其直接前驱元素的值，则操作效率最高的存储结构是（）。
A. 单链表 B. 单循环链表
C. 双向链表 D. 顺序表
4. 假定一个带头结点的链队列的队头和队尾指针分别为 $front$ 和 $rear$ ，则判断队空的条件为（）。
A. $front==rear$ B. $rear!=NULL$
C. $front!=NULL$ D. $front==NULL$
5. 单链表中访问当前结点的直接后继结点的时间复杂度为（）。
A. $O(1)$ B. $O(n)$ C. $O(n^2)$ D. $O(\log n)$
6. 一个非空广义表的表尾（）。
A. 不可能是子表 B. 只能是子表 C. 只能是原子 D. 可以是子表或原子
7. 任何一棵二叉树的叶子结点在中序和后序序列中的相对位置（）。
A. 不发生变化 B. 发生变化 C. 不能确定 D. 以上都不对
8. 在线索化二叉树中，结点 t 的右子树为空的充要条件是（）。
A. $t->rchild==NULL$ B. $t->rtag==1$
C. $t->rtag==1 \&\& t->rchild==NULL$ D. 以上都不对

9. 在一个小顶堆中, 从根结点到某个叶子结点的路径上的结点构成的序列是 ()。
- A. 非递减有序 B. 无序
C. 非递增有序 D. 以上说法均不对
10. 一棵度为 3 的树中, 度为 1 的结点数为 1, 度为 2 的结点数为 2, 度为 3 的结点数为 3, 则度为 0 的结点数为 ()。
- A. 8 B. 9 C. 10 D. 11
11. 若有向图的顶点数为 n , 则该图的弧数最多为 ()。
- A. $n-1$ B. n C. $n*(n-1)/2$ D. $n*(n-1)$
12. 任何一个无向非连通网的最小生成树 ()。
- A. 有一棵或多棵 B. 只有一棵 C. 一定有多棵 D. 可能不存在
13. 利用线性探测法处理冲突的哈希表中, 若将哈希表存储空间看做循环的, 则两个同义词在哈希表中 ()。
- A. 位置可能相邻 B. 位置一定相邻
C. 位置一定不相邻 D. 以上说法均不对
14. 在待排序的元素基本有序的前提下, 效率最高的排序方法是 ()。
- A. 快速排序 B. 直接插入排序 C. 归并排序 D. 简单选择排序
15. 以下排序算法的时间复杂度为 $O(n^2)$ 的是 ()。
- A. 希尔排序 B. 堆排序 C. 归并排序 D. 简单选择排序

二、填空题 (共 20 分, 10 小题, 每题 2 分)

1. 在不带头结点的单链表 L 中, 判空的条件表达式为_____。
2. 队列中, 允许出队的一端称为_____。
3. 已知一棵完全二叉树中共有 701 个结点, 则该树中共有_____个叶子结点。
4. 一棵结点数为 200 的二叉树, 其最小高度为_____。
5. 在二叉查找树中, 插入某个结点后又将其删除, 则所得的二叉查找树_____变化。
6. 平衡二叉树的平衡因子取值只可能是_____。

7. 哈希表属于_____态查找表。

8. 用邻接表存储的具有 n 个顶点和 e 条边的无向图, 其遍历的时间复杂度为_____。

9. 用迪杰斯特拉算法求解带权有向图的最短路径问题时, 要求每条边的权重必须是_____。

10. 对序列 $(27, 21, 15, 18, 41, 7, 12)$ 进行一趟增量为 3 的非递减的希尔排序, 得到的序列为_____。

三、解答题 (共 42 分, 6 小题, 每题 7 分)

1. (7 分) 画出广义表 $((a, b), c)$ 的存储结构。

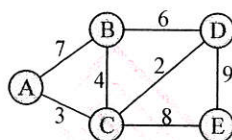
2. (7 分) 从空树开始构造一棵平衡二叉树, 依次插入的关键字为 $(6, 12, 19, 21, 37)$, 请画出该平衡二叉树构造的全过程。

3. (7 分) 已知有向无环图 $G = \langle V, TE \rangle$, 其中 $V = \{A, B, C, D, E, F\}$, $TE = \{ \langle A, B \rangle, \langle A, C \rangle, \langle B, C \rangle, \langle B, D \rangle, \langle C, D \rangle, \langle C, E \rangle, \langle D, F \rangle, \langle E, F \rangle \}$,

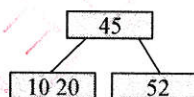
(1) (3 分) 试画出图 G ;

(2) (4 分) 给出 G 的所有可能的拓扑序列。

4. (7 分) 已知如图所示的无向带权图 G , 若采用普里姆 (Prim) 算法构造图 G 的最小生成树, 请给出从 A 顶点出发选取边的顺序。



5. (7 分) 画出在下面的 3 阶 B 树上插入关键字 30 后的 B 树。



6. (7 分) 对序列 $(35, 59, 21, 18, 38, 9, 20)$ 进行非递减的快速排序。若以第一个关键字为枢轴, 请写出该次划分之后的结果。

四、算法阅读题（共 24 分，3 小题，每题 8 分）

1. (8 分) 已知有序表 L1 和 L2，阅读算法 f1，回答下列问题：

(1) (4 分) 若 L1 = (2, 4, 7, 7), L2 = (1, 2, 3, 3, 5), L 为空表，请写出执行算法 f1(L, L1, L2) 后的 L；

(2) (4 分) 简述算法 f1 的功能。

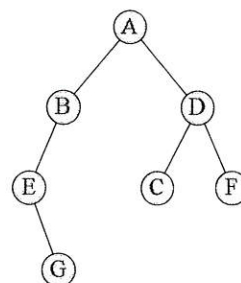
```
void f1(SqList &L, SqList L1, SqList L2){
    int i, j, k;
    i = L1.length-1; j = L2.length-1; k = 0;
    while(i>=0 && j>=0) {
        if(L1.elem[i]>L2.elem[j]) L.elem[k++] = L1.elem[i--];
        else L.elem[k++] = L2.elem[j--];
    }
    while(i>=0) L.elem[k++] = L1.elem[i--];
    while(j>=0) L.elem[k++] = L2.elem[j--];
    L.length = k;
}
```

2. (8 分) 已知二叉树 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，请画出执行算法 f2(T) 后的 T；

(2) (4 分) 简述算法 f2 的功能。

```
void f2(BiTree &T){
    if(T!=NULL) {
        if(NULL==T->lchild && NULL==T->rchild)
            { T = NULL; return; }
        f2(T->lchild);
        f2(T->rchild);
    }
}
```



3. (8 分) 阅读算法 f3，回答下列问题：

(1) (4 分) 若顺序表 L = (45, 34, 78, 12, 29, 64)，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){
    ElemType temp;
    int i, j;
```

```

    for(i = 1; i<L.length; i++){
        j = i;
        while(j>0 && L.elem[j]<L.elem[j-1]) {
            temp = L.elem[j]; L.elem[j] = L.elem[j-1];
            L.elem[j-1] = temp; j--;
        }
    }
}

```

五、算法填空题(共 24 分,3 小题,每题 8 分)

1. (8分) 设链队列的类型定义如下:

```

typedef struct QNode{ // 结点类型
    ElemType data;
    struct QNode *next;
}QNode, *QueuePtr;

typedef struct{ // 链队列类型
    QueuePtr front; // 队头指针
    QueuePtr rear; // 队尾指针
}LinkQueue;

```

算法f4清空带头结点的链队列。请在空缺处填入合适内容,使其成为完整的算法。

```

void f4(LinkQueue &Q) {
    LinkQueue s, p = ①;
    while(p!=NULL) {
        s = p;
        p = ②;
        ③;
    }
    Q.front->next = NULL;
    ④;
}

```

2. (8分) 设二叉查找树 T 采用二叉链表存储结构,其类型定义如下:

```

typedef struct BiTNode{
    TElemType data;

```

```

    struct BiTNode *lchild, *rchild;
}BiTNode, *BiTree;

```

算法 f5 判断二叉查找树 T 是否为平衡二叉树，若是，返回 TRUE，否则返回 FALSE。请在空缺处填入合适内容，使其成为完整的算法。

```

Status f5(BiTree T, int &h){
    int h1, h2;
    if(NULL==T) { ① ; return TRUE;}
    if( ② ){ h = 1; return TRUE;}
    if(!f5(T->lchild, h1)|| ③ )
        return FALSE;
    ④ ;
    if(abs(h1-h2)<2) return TRUE; //abs为求绝对值的函数
    else return FALSE;
}

```

3. (8 分) 设连通图 G 采用邻接表存储结构，其类型定义如下：

```

typedef struct AdjVexNode{
    int adjvex;
    struct AdjVexNode *nextArc;
} AdjVexNode, *AdjVexNodeP; // 邻接链表的结点类型

typedef struct VexNode{
    VexType data;
    struct AdjVexNode *firstArc;
} VexNode; // 顶点数组的元素类型

typedef struct {
    VexNode *vexs; // 顶点数组，用于存储顶点信息
    int n, e; // 顶点数和边（弧）数
    GraphKind kind; // 图的类型
    int *tags; // 标志数组，可用于在图的遍历中标记顶点访问与否
                //G.tags[0..G.n-1]初值为 0，表示所有顶点均未被访问过，
                //当 i 顶点被访问时，其 G.tags[i]的值置为 1
} ALGraph; // 邻接表类型

```

算法 f6 实现从 k 顶点出发对连通图 G 进行深度优先遍历。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 k) {

```

```

AdjVexNodeP p;
printf("%c ", G.vexs[k]);
    ①
for (    ②    ; p!=NULL;    ③    )
    if (0==G.tags[p->adjvex])
        ④
}

```

六、算法设计题 (共10分)

将编号为0和1的两个栈存储在一个一维数据空间 $elem[n]$ 中, 栈底分别位于数组的两端, 其中当0号栈顶指针 $top[0]$ 等于-1时, 该栈为空, 当1号栈顶指针 $top[1]$ 等于 n 时, 该栈为空。两个栈均从两端向中间增长, 如图所示。此双向共享栈的类型定义如下, 请实现栈的初始化、入栈和出栈操作。



```

typedef struct {
    ElemType *elem;
    int top[2];
    int n;
} SqStack;

```

计算机/软件工程专业

每个学校的

考研真题/复试资料/考研经验

考研资讯/报录比/分数线

免费分享



微信 扫一扫

关注微信公众号

计算机与软件考研