

广东工业大学
 2007 年攻读硕士学位研究生入学考试试题

考试科目(代码)名称数据结构

满分: 150

使用专业: 计算机系统结构、. 计算机软件与理论、计算机应用技术

(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

一. 填空题(共 24 分)

- 1 算法的五个重要特性是有穷性, _____, _____, 输入和输出。
2. 链式存储结构的特点是借助_____来表示逻辑关系的。
- 3 栈的修改是按_____的原则进行的。
- 4 与其他线性结构的操作相比. 串操作的特点是_____
- 5 二维数组 A[10][9]采用行优先的存储方法. 若每个元素各占 4 个存储单元 A[0][0]的地址为 500, 则元素 A[3][6]的地址为_____
- 6 在具有 n 个结点的二叉链表中. 共含有_____个空指针域
7. 前序和中序序列相同的二叉树的特征是_____
- 8 具有 n 个顶点的强连通图至少_____条弧
9. 在含有 n 个顶点的连通图中, 任意两个不同顶点之间的一条简单路径的边的数目最多是_____
10. 哈希函数的常用构造方法有_____, _____, 和_____
11. 在一棵 m 阶 B-树 (1>2) 层的结点个数至少是_____
- 12 在读写磁盘上一块信息过程中, 在所谓“等待时间”里是要等待_____

二, 单项选择题

- 13 若算法中的语句最大频度为 $T[n]=n(2007+\log n)+20(\log \uparrow 2)n$, 则其时间复杂度为
 [A] $O(n)$ [B] $O(\log \uparrow 2 n)$ [C] $O(n \log n)$ [D] $O(n \uparrow 2)$
- 14 在长度为 n 的顺序表中删除第 i 个元素的平均复杂度为
 [A] $O(1)$ [B] $O(n)$ [C] $O(i)$ [D] $O(\log n)$
- 15 若一个栈的入栈序列为 1, 2, 3, 4, 5, 6, 则不可能得到_____的出栈序列
 [A] 654321 [B] 354621 [C] 135246 [D] 135642
- 16 模式串 'ababa' 各字符的 next 函数值依次为:
 [A] 01123 [B] 01234 [C] 00123 [D] 11223
- 17 对广义表 $L=((a, b), ((d), (e, f)))$ 执行 $\text{GetHead}(\text{GetTail}(\text{GetHead}(\text{GetTail}(L))))$ 的结果是
 [A] d [B] e [C] (e) [D] (e, f)
- 18 右边的二叉树的结点的后序序列是
 [A] DEBFGCA [B] DBEAFCG
 [C] DBEFCGA [D] ABCDEFG
- 19 如果哈夫曼树 T 有 n 个叶子结点, 则 T 的结点总数为
 [A] $2n$ [B] $2n+1$ [C] $2n-1$ [D] $2(n-1)$
- 20 如果某图的邻接矩阵的上三角部分均为零元, 则该图是
 [A] 完全图 [B] 无环图 [C] 强连通图 [D] 非连通图
- 21 由 250 个关键字逐个插入生成的二叉排序树的最低深度为

[A] 8 [B] 9 [C] 10 [D] 11

22 深度为 5 的平衡二叉树的结点个数至少是

[A] 12 [B] 13 [C] 15 [D] 16

23 对关键字序列(5, 1, 4, 3, 7, 2, 8, 6)进行快速排序时,以第一个元素 5 为枢轴元素的一次划分的结果为

[A] (1, 2, 3, 4, 5, 6, 7, 8) [B] (2, 1, 4, 3, 5, 7, 8, 6)
[C] (1, 4, 3, 2, 5, 7, 8, 6) [D] (8, 7, 6, 5, 4, 3, 2, 1)

24 稀疏索引是在索引表中

[A] 为每个字段设一个索引项 [B] 为每个记录设一个索引项
[C] 为每组字段设一个索引项 [D] 为每组记录设一个索引项

三. 解答题(共 45 分)

25. (10 分)画出以下稀疏矩阵的下列存储结构

0	0	0	0	0
6	0	5	3	0
0	0	0	0	8
0	0	7	5	0

(1)三元组表;
(2)十字链表.

26. (6 分)已知一棵二叉树的中序序列和后续序列分别为

BFCDEA
FBDEAC

请画出该二叉树.

27. (9 分)对序列(29, 14, 22, 40, 55, 37)执行升序的堆排序算法,请以二叉树的形式画出初始堆和排序过程中各趟的结果.

28 (12 分) 已知带权图 G 如下

画出图 G 的邻接表存储结构
画出图 G 的一棵最小生成树

29 (8 分) 设有一棵线索化的二叉排序树如右图所示

请问该树是按照哪一种遍历次序线索化的
画出插入结点 D 并修改相应线索后的该树

四. 算法题(共 57 分)

30. (7 分) 假设某个单向循环链表的长度大于 1, 且该表中既无头结点也无头指针. s 为指向链表中某个结点的指针. 算法 f30 在链表中删除指针 s 所指结点的前驱结点. 请在空缺处填入合适内容, 使其成为完整的算法.

```
ElemType f30(LinkList s) {
    LinkList pre, p;
    ElemType e;
    pre=s; p=s->next;
    while ( ① ) {
        pre=p;
        p=( ② );
    }
    pre->next=( ③ );
    e=p->data;
    free ( ④ );
    Return e;
}
```

31. (7 分) L 是一个带头结点的递增有序的单链表. 阅读算法 f31, 并回答问题:

- (1) 设 L 中的元素值依次为 (1, 2, 2, 3, 3, 3, 4, 4, 4, 4), 请写出执行算法 f31 (L) 后 L 的元素值;
- (2) 简述算法 f31 的功能.

```
void f31(LinkList &L) {
    LinkList p, q, s;
    p=L->next;
    while(p!=NULL) {
        q=p->next;
        while(q!=NULL 且 q->data==p->data)
            {s=q->next; free (q);q=s; }
        p->next=q; p=q;
    }
}
```

32. (7 分) 假设以两个带头结点的递增有序单链表 ha 和 hb 分别表示两个集合. 阅读算法 f32 回答问题:

- (1) 若两个集合分别为: (3, 5, 7, 9) 和 (2, 3, 4, 5, 6). 并分别存储在表 A 和 B 中. 请写出执行算法 f32 (A, B) 后. 表 A 和表 B 各自表示的集合;
- (2) 简述算法 f32 的功能.

```
void f32(LinkList &ha, LinkList &hb) {
    LinkList pa, pb, q, r, t;
    pa=ha->next; pb=hb->next;
    r=ha; t=hb;
```

```

while(pa!=NULL && pb!=NULL) {
    if(pa->data < pb->data)
        { r->next=pa; r=pa; pa=pa->next; }
    else if (pa->data > pb->data)
        { r->next=pb; r=pb; pb=pb->next; }
    else
    {
        q=pa; pa=pa->next; free(q);
        t->next=pb; t=pb; pb=pb->next;
    }
}
r->next=NULL; t->next=NULL;
if(pa) r->next=pa;
if(pb) r->next=pb;
)

```

33. (7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 f33(T) 判定二叉树 T 是否为正则二叉树。请在空缺处填入合适内容, 使其成为完整的算法。

```

int f33(BiTree T) {
    BiTree p; Queue Q;
    if(_____①_____ ) return 1;
    Init Queue (Q); EnQueue(Q, T);
    do{
        DeQueue(Q, p);
        if( p->lchild || p->rchild )
            if(_____②_____ ) {
                EnQueue(Q, P->lchild);
                _____③_____ ;
            } else _____④_____ ;
    } while( !QueueEmpty(Q) );
    return 1;
}

```

34. (7分) 以下函数 f34(T) 求一棵以孩子—兄弟链表表示的树 T 的度。请在空缺处填入合适内容, 使其成为完整的算法。

```

typedef struct CSNode{
    ElemType data;
    struct CSNode *firstchild, *nextsibling; } *CSTree; //树的二叉链表(孩子—兄弟)存储表示
int f34(CSTree T) {
    int d, ds, dt; CSTree p;
    if (!T) return ① ;
    else{ ds=0; dt=0;
    for (p=T->firstchild; p; p=p->nextsibling) {
        dt++;
        d=f34 ( ② ) ;
        if ( ③ ) ds=d;
    }
}

```

```

}
return ds>dt ?( ④ );
} )
    
```

35. (7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; // 顶点的信息
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 f35 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 f35(ALGraph SG, 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 ( ③ );
G.arcNum = _____④_____ ;
}
}
}
    
```

36. (15 分) 某仓库用一个带头结点的有序单链表 A 存储各种货物的代码和数量, 并用另一个类型相同的有序单链表 B 存储客户的订货信息 (两个链表均已按代码从小到大排序)。链表的类型定义如下:

```
typedef struct Goods {  
    int    code; // 代码  
    int    total; // 代码  
    Goods *next;  
} Goods, *GoodsList;
```

(1) 写一算法, 根据订货表 B 检查库存表 A, 产生一个缺货表 C, 其中表 C 的结点登记缺货的货物的代码和缺货数量。例如, 如果有下表 A 和 B, 则应产生相应的表 C (注意, 表 A 和 B 不改变):

(2) 给出你的算法的时间复杂度