

广东工业大学
2004 年攻读硕士学位研究生入学考试试题

考试科目：数据结构

招生专业：

科目代码：418

(注：试卷封面需填写自己的准考证号，答完后连同试题一并交回。)

一. 填空题(共 24 分)

1. 构成抽象数据类型的三个要素是_____，_____和_____。
2. 顺序存储结构的特点是借助_____表示数据元素之间的逻辑关系。
3. 循环链表的主要优点是_____。
4. 无表头结点的链队列 Q 为空的条件是_____。
5. 若模式串 $p = 'xyzxyx'$ ，则 p 的 next 函数值依次为_____。
6. 设二维数组 $A[20][10]$ 按列优先顺序存储，每个元素占 4 个单元， $A[0][0]$ 的存储地址是 500，则 $A[13][8]$ 的存储地址是_____。
7. 只有一个结点的二叉树的高度为 1，高度为 k 的二叉树至少有_____个结点。
8. 如果用邻接矩阵存储具有 n 个顶点的图，则对该图进行深度优先遍历的算法时间复杂度为_____。
9. 在一棵 m 阶 B. 树中，第 $(i>1)$ 层的结点个数至少是_____。
10. 用某种排序方法对序列(24, 83, 20, 46, 17, 31, 66, 35, 18)进行排序时，前 3 趟结果如下
(35, 83, 66, 46, 31, 24, 17, 20, 18)
(46, 83, 66, 35, 31, 24, 18, 20, 17)
(66, 83, 46, 35, 31, 24, 20, 18, 17)
则所采用的排序方法为_____。

二. 单项选择题(共 24 分)

13. 语句的频度是指在程序中该语句的_____ J
[A] 执行次数 [B] 出现次数 [C] 循环嵌套层数 [D] 条件嵌套层数
14. 对于一个头指针为 head 的带头结点的循环链表，判定该表为空表的条件是
[A] $head == NULL$ [B] $head \rightarrow next == NULL$
[C] $head != NULL$ [D] $head \rightarrow next == head$
15. 若以 a, b, c, d 作为输入序列，则能由输出受限的双端队列得到，但不能由输入受限的双端队列得到的输出序列是
[A] a, b, C, d [B] d, C, b, a [C] b, a, d, C [D] d, b, a, c
16. 以下陈述中正确的是
[A] 空串只含空格 [B] 串只含字母和数字
[C] 串是一种特殊的线性表 [D] 串的长度大于零
17. 广义表 $A = (a, b, (C, d), (e, (f, g)))$ ， $Head(Tail(Head(Tail(Tail(A)))))$ 的值为
[A] C [B] d [C] (d) [D] (g)
18. 已知-A 树的前序序列为 ABDCEFG，中序序列为 DBCAFEG，则其后序序列为
[A] DCBAFGE [B] DCBFGEA [C] DCBFEGA [D] DCBGFEA

19. 一个含有 n 个顶点和 n 条弧的强连通图是
 [A]无回路的 [B]树状的 [c]环状的 [D]重连通的
20. 在散列函数 $H(k)=k\%p$ 中, P 应为
 [A]奇数 [B]偶数 [c]充分大的数 [D]素数
21. 下列排序方法中, 时间复杂度与元素初始排列无关的是
 [A]快速排序 [B]直接插入排序 [c]堆排序 [D]直接选择排序
22. 根据堆的定义, 下面四个序列中, 不构成堆的是
 [A](1, 2, 3, 4, 10, 9, 8, 7, 6, 5) [B](10, 5, 8, 4, 2, 6, 7, 1, 3)
 [C](10, 9, 8, 7, 3, 5, 4, 6, 2) [D](1, 2, 5, 3, 4, 6, 7, 8, 9, 10)
23. 为了提高外部排序的效率, 在进行 k 路归并时应采用
 [A]败者树 [B]胜者树 [C]B-树 [D]键树
24. 散列文件的基本存储单位是
 [A]物理记录 [B]逻辑记录 [C]页块 [D]桶

三. 解答题(共 45 分)

25. (10 分) 画出下列广义表的共享存储结构(结点结构如图所示)。

(1) $A=(d, ())$

(2) $B=(A, (e, (f)))$

(3) $C=(A, C)$

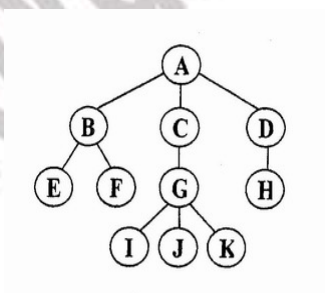
表结点

1	hp
---	----

原子结点

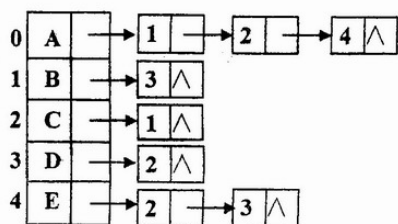
0	
---	--

26. (10 分)给定树 T 如图所示。



- (1) 画出 T 的带双亲的孩子链表;
 (2) 画出 T 对应的二叉树;
 (3) 写出 T 的后序遍历序列。

27. (10 分)给定图 G 的邻接矩阵如图所示。

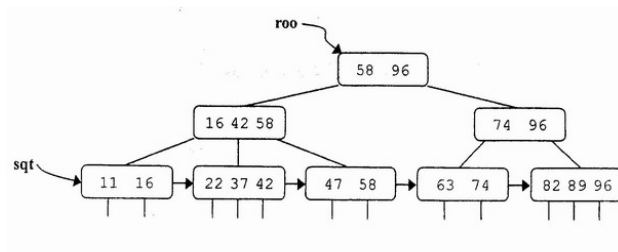


- (1)画出图 G 的深度优先生成森林;
 (2)画出图 G 的广度优先生成森林;
 (3)画出图 G 的全部强连通分量。

28. (10 分)现有一棵 3 阶 B+树如图所示。分别画出依次删除以下关键字之后的 B+树。

(1) 96;

(2) 47;



(3) 63。

29. (5 分)对序列(48, 37, 63, 96, 22, 31, 50, 55, 11)执行升序的希尔排序算法, 增量序列为(5, 3, 1), 写出排序中各趟的结果。

四.算法题 (共 57 分)

30. (7 分) 阅读下列排序算法,并回答问题

(1) 设线性表 $La=(2, 4, 6)$ 和 $Lb=(1, 3, 5, 7)$, 并采用 带头结点的单链表作为存储结构,写出执行 $f30(La, Lb, Lc)$ 之后的 Lc ;

(2) 对于一般的 $La=(a_1, a_2, A, a_m)$ 和 $Lb=(b_1, b_2, A, b_n)$, 简要说明算法 $f30(La, Lb, Lc)$ 的意义, 并给出执行 $f30(La, Lb, Lc)$ 之后的 Lc 的一般形式。

```
void f30(LinkList La, LinkList Lb, LinkList &Lc) {
```

```
    pa=La->next;
```

```
    pb=Lb->next;
```

```
    Lc=La;    pc=Lc;
```

```
    while (pa && pb)    {
```

```
        pc->next=pa;    pc=pa;    pa=pa->next;
```

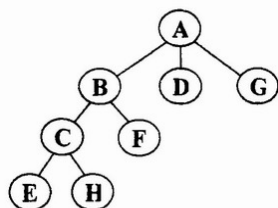
```
        pc->next=pb;    pc=pb;    pb=pb->next;
```

```
    }
```

```
    if (pa)    pc->next=pa;    else    pc->next=pb;
```

```
}
```

31. (7分)如图所示的数 T 采用孩子兄弟链表存储结构,

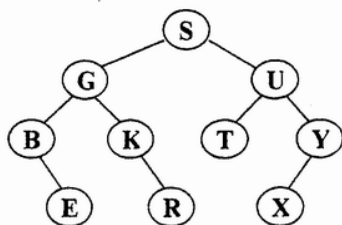


- (1) 请写出调用算法 f31(T) 的返回值;
(2) 简述对任意 T, 算法 f31(T) 的返回值的意义。

```

int f31(CSTree T) {
    if (!T->firstchild) return 1;
    else {
        count=0;
        for (p=T->firstchild; p; p=p->nextsibling)
            count += f31(p);
        return count;
    }
}
  
```

32. (7分) 如图所示的二叉树 t 采用二叉链表存储结构,



- (1) 依次写出调用算法 f32(t, 'K') 的输出;
(2) 对任意二叉排序树 t 和字符 c , 简述算法 f32(t,c) 的意义。

```

void f32(BiTree t, char c) {
    if (t) {
        f32(t->rchild, c);
        if (t->data >= c) {
            printf("%c", t->data);
        }
        f32(t->lchild, c);
    }
}
  
```

33. (7分) 二叉树的存储结构的类型定义如下:

```

typedef struct BiTNode {
  
```

```
char    data;
BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
```

以下函数 f33(T) 对二叉树 T 中所有结点执行如下操作:如果结点没有左孩子但有右孩子,就将其右孩子改变为左孩子.请在空缺处填入合适内容,使其成为完整的算法。

```
void f33(BiTree T) {
if ( _____①_____ ) {
    _____②_____;
f33( T->rchild);
if( _____③_____ ) {
T->lchild = T->rchild;
T->rchild = _____④_____;
}
}
}
```

34. (7 分) 以下非递归算法 f43(n) 求递归函数

的值。请在空缺处填入合适内容, 使其成为完整的算法。

```
Int f34(int n) {
If(n<0) _____①_____;
Else{
S=_____②_____;
While (n>_____③_____) {
S*=n;
_____④_____;
}
}
return s;
}
```

35. (7 分) 已知有向图的邻接表表示的形式描述如下:

```
#define MaxNum 20 //图的最大顶点数
typedef struct ArcNode {
int adjvex; //邻接点域
struct Arc *nextArc; //弧表链指针域
} ArcNode; //弧结点类型
typedef struct {
char vertex; //顶点域
ArcNode *firstArc; //弧表头类型
} VertexNode; //顶点表结点类型
typedef struct {
VertexNode vertices [MaxNum] ; // 顶点表
int vexnum, arcnum; // 图中当前的顶点数和弧数
```



```
} ALGraph; // 邻接表类型
```

如果在有向无环图 G 中存在一个顶点 r , 且存在从 r 到 G 中的其余顶点的有向途径, 则称 r 为 G 的根。请在下列算法的空缺处填入合适内容, 使算法 f35 (G, r) 成为求有向无环图 G 的根 r 的完整算法。

```
Status DFS(ALGraph G, int v, int ficount) {
visited[v]=1;
    ① _____;
for (p=G.vertices[v] .firstArc; p; p=p->nextArc) {
w = p->adjvex;
if (visited [w]==_____②_____) return FALSE;
else if (visited[w]==0)
if (!DFS(G,w,count)) return FALSE;
}
visited [_____③_____] = 2;
return TRUE;
}

Status f35(ALGraph G, int &r) {
// 判定有向图 G 是否有根.若有,则用 r 返回根的序号,且函数值为 TRUE。
FindInDegree(G,indegree); // 求 G 的各顶点入度到数组 indegree[ ] count=0;
for (v=0; v<G.vexnum; v++)
if (indegree[v]==0) { r=v; count++; }
if (count != 1) return FALSE;
else {
for (v=0; v<G.vexnum; v++) visited[v]=0;
count=0;
DFS(G,r,count);
return _____④_____;
}
}
```

36. (15 分) 二叉树 T 采用第 33 题的二叉链表 BiTree 作为存储结构, 写出一个函数 f36 (T), 如果 T 是平衡二叉树 (注: 结点中未设平衡因子域), 则返回 T 的高度, 否则返回 T 的高度负值。应在函数的首部以注释的形式简述算法的基本思想和时间复杂度。