

全国 2001 年 10 月高等教育自学考试

数据结构试题

课程代码： 02331

第一部分 选择题 (30 分)

一、单项选择题 (本大题共 15 小题, 每小题 2 分, 共 30 分) 在每小题列出的四个选项中只有一个选项是符合题目要求的, 请将正确选项前的字母填在题后的括号内。

1. 算法指的是 ()
A. 计算机程序
B. 解决问题的计算方法
C. 排序算法
D. 解决问题的有限运算序列
2. 线性表采用链式存储时, 结点的存储地址 ()
A. 必须是不连续的
B. 连续与否均可
C. 必须是连续的
D. 和头结点的存储地址相连续
3. 将长度为 n 的单链表链接在长度为 m 的单链表之后的算法的时间复杂度为 ()
A. $O(1)$ B. $O(n)$ C. $O(m)$ D. $O(m+n)$
4. 由两个栈共享一个向量空间的好处是: ()
A. 减少存取时间, 降低下溢发生的机率
B. 节省存储空间, 降低上溢发生的机率
C. 减少存取时间, 降低上溢发生的机率
D. 节省存储空间, 降低下溢发生的机率
5. 设数组 $data[m]$ 作为循环队列 SQ 的存储空间, $front$ 为队头指针, $rear$ 为队尾指针, 则执行出队操作后其头指针 $front$ 值为 ()
A. $front=front+1$ B. $front=(front+1)\%(m-1)$
C. $front=(front-1)\%m$ D. $front=(front+1)\%m$
6. 如下陈述中正确的是 ()
A. 串是一种特殊的线性表
B. 串的长度必须大于零
C. 串中元素只能是字母
D. 空串就是空白串
7. 若目标串的长度为 n , 模式串的长度为 $[n/3]$, 则执行模式匹配算法时, 在最坏情况下的时间复杂度是 ()
A. $O(\frac{n}{3})$ B. $O(n)$ C. $O(n^2)$ D. $O(n^3)$
8. 一个非空广义表的表头 ()
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
9. 假设以带行表的三元组表表示稀疏矩阵, 则和下列行表

0	2	3	3	5
---	---	---	---	---

对应的稀疏矩阵是 ()

A.
$$\begin{bmatrix} 0 & -8 & 0 & 6 \\ 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

B.
$$\begin{bmatrix} 0 & -8 & 0 & 6 \\ 7 & 0 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{bmatrix}$$

C.
$$\begin{bmatrix} 0 & -8 & 0 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

D.
$$\begin{bmatrix} 0 & -8 & 0 & 6 \\ 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 3 & 0 & 0 \end{bmatrix}$$

10. 在一棵度为 3 的树中,度为 3 的结点个数为 2,度为 2 的结点个数为 1,则度为 0 的结点个数为 ()

- A. 4 B. 5 C. 6 D. 7

11. 在含 n 个顶点和 e 条边的无向图的邻接矩阵中,零元素的个数为 ()

- A. e B. $2e$ C. $n^2 - e$ D. $n^2 - 2e$

12. 假设一个有 n 个顶点和 e 条弧的有向图用邻接表表示,则删除与某个顶点 v_i 相关的所有弧的时间复杂度是 ()

- A. $O(n)$ B. $O(e)$ C. $O(n+e)$ D. $O(n \cdot e)$

13. 用某种排序方法对关键字序列 (25, 84, 21, 47, 15, 27, 68, 35, 20) 进行排序时,序列的变化情况如下:

20, 15, 21, 25, 47, 27, 68, 35, 84

15, 20, 21, 25, 35, 27, 47, 68, 84

15, 20, 21, 25, 27, 35, 47, 68, 84

则所采用的排序方法是 ()

- A. 选择排序 B. 希尔排序 C. 归并排序 D. 快速排序

14. 适于对动态查找表进行高效率查找的组织结构是 ()

- A. 有序表 B. 分块有序表 C. 三叉排序树 D. 线性链表

15. 不定长文件是指 ()

- A. 文件的长度不固定 B. 记录的长度不固定
C. 字段的长度不固定 D. 关键字项的长度不固定

第二部分 非选择题 (共 70 分)

二、填空题 (本大题共 10 小题,每小题 2 分,若有两个空格,每个空格 1 分,共 20 分) 不写解答过程,将正确的答案写在每小题的空格内。错填或不填均无分。

16. 数据的逻辑结构是从逻辑关系上描述数据, 它与数据的 _____ 无关, 是独立于计算机的。

17. 在一个带头结点的单循环链表中, p 指向尾结点的直接前驱, 则指向头结点的指针 $head$ 可用 p 表示为 $head = \underline{\hspace{2cm}}$ 。

18. 栈顶的位置是随着 _____ 操作而变化的。

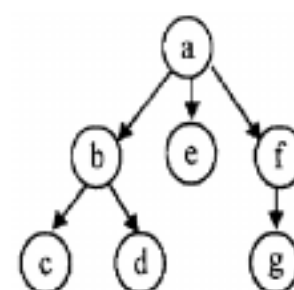
19. 在串 $S = \text{"structure"}$ 中,以 t 为首字符的子串有 _____ 个。

20. 假设一个 9 阶的上三角矩阵 A 按列优先顺序压缩存储在一维数组 B 中,其中 $B[0]$ 存储

矩阵中第 1 个元素 $a_{1,1}$, 则 $B[31]$ 中存放的元素是 _____。

21. 已知一棵完全二叉树中共有 768 结点, 则该树中共有 _____ 个叶子结点。

22. 已知一个图的广度优先生成树如右图所示, 则与此相应的广度优先遍历序列为 _____。



23. 在单链表上难以实现的排序方法有 _____ 和 _____。

24. 在有序表 (12, 24, 36, 48, 60, 72, 84) 中二分查找关键字 72 时所需进行的关键字比较次数为 _____。

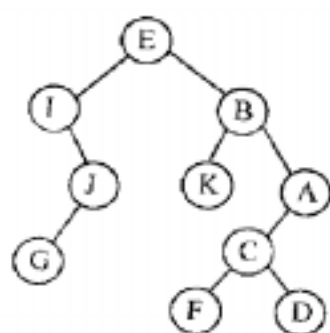
25. 多重表文件和倒排文件都归属于 _____ 文件。

三、解答题 (本大题共 4 小题, 每小题 5 分, 共 20 分)

26. 画出下列广义表的共享结构图形表示

$P = (((z), (x, y)), ((x, y), x), (z))$

27. 请画出与下列二叉树对应的森林。



28. 已知一个无向图的顶点集为 {a, b, c, d, e}, 其邻接矩阵如下所示

a	0	1	0	0	1
b	1	0	0	1	0
c	0	0	0	1	1
d	0	1	1	0	1
e	1	0	1	1	0

(1) 画出该图的图形;

(2) 根据邻接矩阵从顶点 a 出发进行深度优先遍历和广度优先遍历, 写出相应的遍历序列。

29. 已知一个散列表如下图所示:

		35		20			33		48			59
0	1	2	3	4	5	6	7	8	9	10	11	12

其散列函数为 $h(\text{key}) = \text{key} \% 13$, 处理冲突的方法为双重散列法, 探查序列为:

$$h_i = (h(\text{key}) + i * h_1(\text{key})) \% m \quad i = 0, 1, \dots, m - 1$$

其中

$$h_1(\text{key}) = \text{key} \% 11 + 1$$

回答下列问题:

(1) 对表中关键字 35, 20, 33 和 48 进行查找时, 所需进行的比较次数各为多少?

(2) 该散列表在等概率查找时查找成功的平均查找长度为多少?

四、算法阅读题 (本大题共 4 小题, 每小题 5 分, 共 20 分)

30 . 下列算法的功能是比较两个链串的大小，其返回值为：

$$\text{comstr}(s_1, s_2) = \begin{cases} -1 & \text{当 } s_1 < s_2 \\ 0 & \text{当 } s_1 = s_2 \\ 1 & \text{当 } s_1 > s_2 \end{cases}$$

请在空白处填入适当的内容。

```
int comstr(LinkString s1, LinkString s2)
{//s1 和 s2 为两个链串的头指针
    while(s1 && s2){
        if(s1 -> date < s2 -> date) return -1 ;
        if(s1 -> date > s2 -> date) return 1 ;
        _____ ;
        _____ ;
    }
    if( _____ ) return -1 ;
    if( _____ ) return 1 ;
    _____ ;
}
```

31 . 阅读下面的算法

```
LinkList mynote(LinkList L)
{//L 是不带头结点的单链表的头指针
    if(L && L->next){
        q=L ; L=L ->next ; p=L ;
        S1 :      while(p ->next) p=p ->next ;
        S2 :      p ->next=q ; q ->next=NULL ;
    }
    return L ;
}
```

请回答下列问题：

(1) 说明语句 S1 的功能；

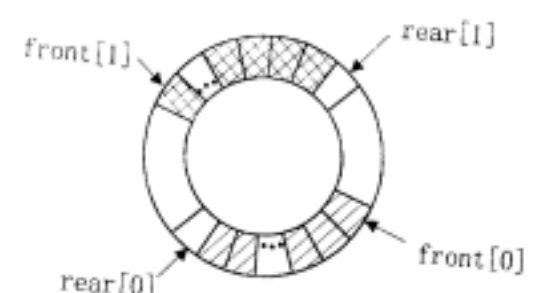
(2) 说明语句组 S2 的功能；

(3) 设链表表示的线性表为 (a_1, a_2, \dots, a_n), 写出算法执行后的返回值所表示的线性表。

32 . 假设两个队列共享一个循环向量空间 (参见右下图)

其类型 Queue2 定义如下：

```
typedef struct{
    DateType data[MaxSize] ;
```



```

        int front[2],rear[2] ;
    }Queue2 ;
    对于 i=0 或 1 ,front[i] 和 rear[i] 分别为第 i 个队列的头指针和尾指针。 请对以下算法填空 ,
    实现第 i 个队列的入队操作。

```

```

    int EnQueue (Queue2*Q,int i,DataType x)
    {
        // 若第 i 个队列不满,则元素 x 入队列,并返回 1;否则返回 0
        if(i<0||i>1)return 0 ;
        if(Q ->rear[i]==Q ->front[ _____]return 0 ;
        Q ->data[ _____]=x ;
        Q ->rear[i]=[ _____];
        return 1 ;
    }

```

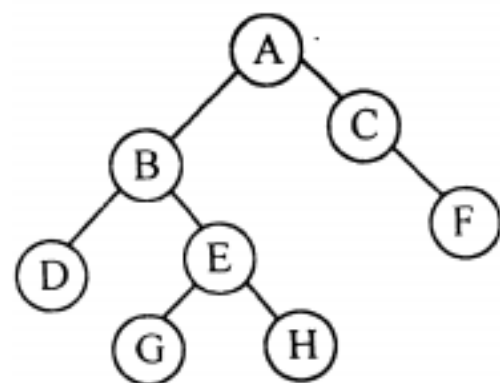
33 . 已知二叉树的存储结构为二叉链表,阅读下面算法。

```

typedef struct node {
    DataType data ;
    Struct node * next ;
}ListNode ;
typedef ListNode * LinkList ;
LinkList Leafhead=NULL ;
Void Inorder (BinTree T)
{
    LinkList s ;
    If(T){
        Inorder(T ->lchild) ;
        If ((!T ->lchild)&&(!T ->rchild)){
            s=(ListNode*)malloc(sizeof(ListNode)) ;
            s ->data=T ->data ;
            s ->next=Leafhead ;
            Leafhead=s ;
        }
        Inorder(T ->rchild) ;
    }
}

```

对于如下所示的二叉树



(1) 画出执行上述算法后所建立的结构；

(2) 说明该算法的功能。

五、算法设计题 (本题共 10 分)

34 . 阅读下列函数 arrange()

```
int arrange(int a[],int l,int h,int x)
{//l 和 h 分别为数据区的下界和上界
    int i,j,t ;
    i=l ; j=h ;
    while(i<j){
        while(i<j && a[j]>=x)j-- ;
        while(i<j && a[i]<=x)i++ ;
        if(i<j)
            { t=a[j] ; a[j]=a[i] ; a[i]=t ; }
    }
    if(a[i]<x) return i ;
    else return i - 1 ;
}
```

(1) 写出该函数的功能；

(2) 写一个调用上述函数实现下列功能的算法： 对一整型数组 b[n] 中的元素进行重新排列，将所有负数均调整到数组的低下标端， 将所有正数均调整到数组的高下标端， 若有零值，则置于两者之间，并返回数组中零元素的个数。

全国 2001 年 10 月高等教育自学考试

数据结构试题参考答案

课程代码： 02331

一、单项选择题 (本大题共 15 小题，每小题 2 分，共 30 分)

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

二、填空题 (本大题共 10 小题，每小题 2 分，共 20 分)

16 . 存储 (或存储结构) 17.p - > next - > next
18 . 进栈和退栈 19 . 12
20 . $a_{4,8}$ 21 . 384
22 . abefcdg

23 . 快速排序、堆排序、希尔排序

24 . 2

25.多关键字

三、解答题 (本大题共 4 小题，每小题 5 分，共 20 分)

26 .

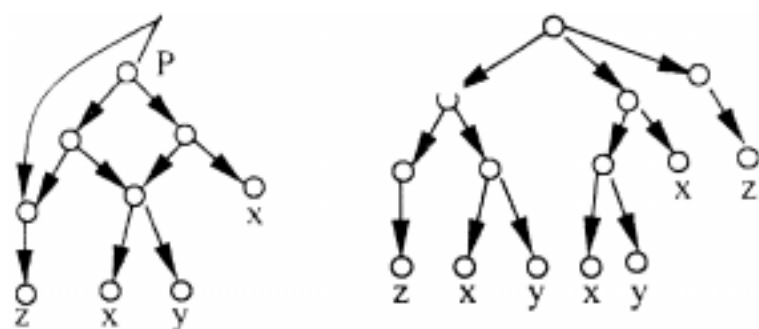
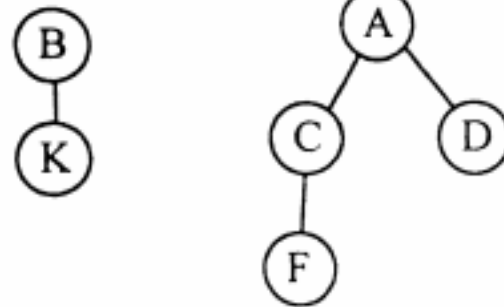
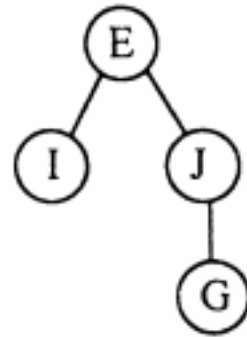


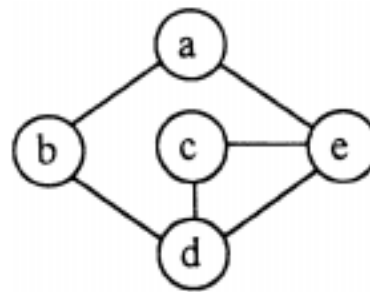
图 1

图 2

27 .



28 . 该图的图形为：



深度优先遍历序列为： abdce

广度优先遍历序列为： abedc

29 . (1) 对关键字 35、20、33 和 48 进行查找的比较次数为 3、2、1、1；

$$(2) \text{ 平均查找长度 } ASL = \frac{3+2+1+1+2}{5} = \frac{9}{5}$$

四、算法阅读题（本大题共 4 小题，每小题 5 分，共 20 分）

30 .
 $S1 = S1 \rightarrow next$
 $s2 = s2 \rightarrow next$
 $s2$ (或 $s2 \neq NULL$ 或 $s2 \&\& !s1$)
 $s1$ (或 $s1 \neq NULL$ 或 $s1 \&\& !s2$)
 return 0

31. (1) 查询链表的尾结点
 (2) 将第一个结点链接到链表的尾部，作为新的尾结点
 (3) 返回的线性表为 ($a_2, a_3, \dots, a_n, a_1$)

32. $(i + 1) \% 2$ (或 $1 - i$)
 $Q \rightarrow rear[i]$
 $(Q \rightarrow rear[i] + 1) \% Maxsize$

33.(1) Leafhead \longrightarrow

F	\longrightarrow
---	-------------------

H	\longrightarrow
---	-------------------

G	\longrightarrow
---	-------------------

D	\longrightarrow
---	-------------------

(2) 中序遍历二叉树，按遍历序列中叶子结点数据域的值构建一个以 Leafhead 为头指针的逆序单链表（或按二叉树中叶子结点数据自右至左链接成一个链表）。

五、算法设计题（本题共 10 分）

34 . (1) 该函数的功能是：调整整数数组 $a[]$ 中的元素并返回分界值 i ，使所有 $< x$ 的元素均落在 $a[1..i]$ 上，使所有 $> x$ 的元素均落在 $a[i + 1..h]$ 上。

(2) int f(int b[],int n)	或	int f(int b[],int n)
{		{
int p,q ;		int p,q ;
p=arrange(b,0,n - 1,0) ;		p=arrange(b,0,n - 1,1) ;
q= arrange(b,p+1,n - 1,1) ;		q= arrange(b,0,p,0) ;
return q - p ;		return p - q ;
}		}

2003.1 数据结构试题

一、单项选择题 (本大题共 15 小题，每小题 2 分，共 30 分。在每小题的四个备选答案中，选出一个正确答案，并将正确答案的序号填在题干的括号内)

1.下面程序段的时间复杂度是 (D)

```
for(i=0;i<n;i++)
    for(j=1;j<m;j++)
        A[i][j]=0 ;
```

A.O(n) B.O(m+n+1) C.O(m+n) **D.O(m*n)**

2.在单链表中，指针 p 指向元素为 x 的结点，实现“删除 x 的后继”的语句是 (B)

A.p=p->next; **B.p->next=p->next->next;**
C.p->next=p; D.p=p->next->next;

3.在头指针为 head 且表长大于 1 的单循环链表中，指针 p 指向表中某个结点，若 p->next->next= head,则(D)

A.p 指向头结点 B.p 指向尾结点
C.*p 的直接后继是头结点 **D.*P 的直接后继是尾结点**

4.判定“带头结点的链队列为空”的条件是 (C)

A.Q.front==NULL B.Q.rear==NULL
C.Q.front==Q.rear D.Q.front!=Q.rear

5.设有两个串 T 和 P，求 P 在 T 中首次出现的位置的串运算称作 (D)

A.联接 B.求子串 C.字符定位 **D.子串定位**

6.广义表 A=(a,(b),(c,d,e)) 的长度为 (A)

A.4 B.5 C.6 **D.7**

7.一棵含 18 个结点的二叉树的高度至少为 (C)

A.3 B.4 **C.5** D.6

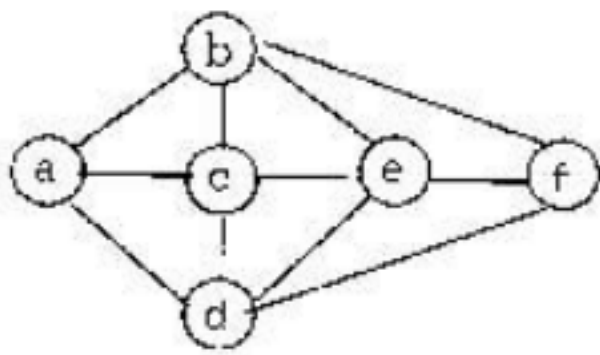
8.已知二叉树的先序序列为 ABDECF，中序序列为 DBEAFC，则后序序列为 (D)

A.DEBAFC B.DEFBCA C.DEBCFA **D.DEBFCA**

9.无向图中一个顶点的度是指图中 (B)

A.通过该顶点的简单路径数 **B.与该顶点相邻接的顶点数**
C.通过该顶点的回路数 D.与该顶点连通的顶点数

10.已知一个图如下所示，从顶点 a 出发进行广度优先遍历可能得到的序列为 (C)



题 10 图

A.acefb d B.acb dfe C.a**c b d e f** D.acdbfe

11.在下列排序方法中，平均时间性能为 $O(n\log n)$ 且空间性能最好的是 (B)

A.快速排序 B.堆排序 C.归并排序 D.基数排序

12.已知一组关键字为 {25,48,36,72,79,82,23,40,16,35}，其中每相邻两个为有序子序列。对这些子序列进行一趟两两归并的结果是 (A)

A.{**25,36,48,72,23,40,79,82,16,35**}

B.{25,36,48,72,16,23,40,79,82,35}

C.{25,36,48,72,16,23,35,40,79,82}

D.{16,23,25,35,36,40,48,72,79,82}

13.设顺序存储的线性表共有 123 个元素，按分块查找的要求等分成 3 块。若对索引表采用顺序查找来确定块，并在确定的块中进行顺序查找，则在查找概率相等的情况下，分块查找成功时的平均查找长度为 (B)

A.21

B.**23**

C.41

D.62

14.索引非顺序文件的特点是 (A)

A.主文件无序，索引表有序

B.主文件有序，索引表无序

C.主文件有序，索引表有序

D.主文件无序，索引表无序

15.倒排文件的主要优点是 (C)

A.便于进行插入和删除运算

B.便于进行文件的恢复

C.便于进行多关键字查询

D.节省存储空间

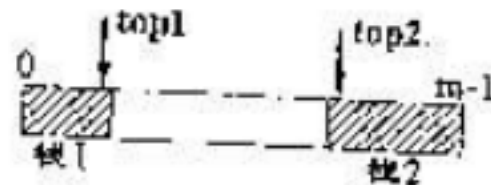
二、填空题 (本大题共 10 小题，每小题 2 分，若有两个空格，每个空格 1 分，共 20 分)

16.抽象数据类型的特点是将 ___数据___和___运算___封装在一起，从而现实信息隐藏。

17.从顺序表中删除一个元素时，表中所有在被删元素之后的元素均需 ___前移___一个位置。

18.在队列中，允许进行插入操作的一端称为 ___队尾___，允许进行删除操作的一端称为 ___队头___。

19.如图两个栈共享一个向量空间，top1 和 top2 分别为指向两个栈顶元素的指针，则“栈满”



题 19 图

的判定条件是 top1=top2-1。

20.设 $S1=\text{"good"}$, $S2=\text{" "}$, $S3=\text{"book"}$ ，则 $S1$, $S2$ 和 $S3$ 依次联接后的结果是 good book。

21.假设三维数组 $A[10][9][8]$ 按行优先顺序存储，若每个元素占 3 个存储单元，且首地址为

100，则元素 $A[9][8][7]$ 的存储地址是 2257。

22.已知在一棵含有 n 个结点的树中，只有度为 k 的分支结点和度为 0 的叶子结点，则该树中含有的叶子结点的数目为 $(n-1)/k + 1$ 或 $n - (n-1)/k$ 。

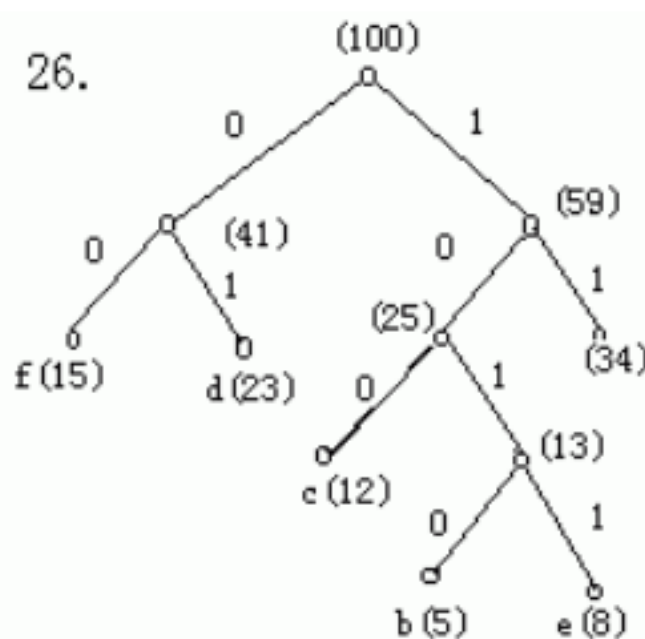
23.能够成功完全拓扑排序的图一定是一个 有向无环图。

24.如果在排序前，关键字序列已接近正序或逆序，则在堆排序和快速排序两者之中，选用 堆排序 较为适当。

25.假设哈希表的表长为 m ，哈希函数为 $H(key)$ ，若用线性探查法解决冲突，则探查地址序列的形式表达为 $hi = (H(key) + i) / m$ 。

三、解答题（本大题共 4 小题，每小题 5 分，共 20 分）

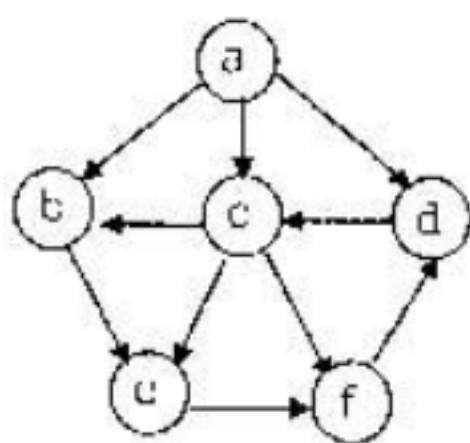
26.假设通信电文使用的字符集为 $\{a, b, c, d, e, f\}$ ，各字符在电文中出现的频度分别为：34, 5, 12, 23, 8, 18，试为这 6 个字符设计哈夫曼编码。请先画出你所构造的哈夫曼树（要求树中左孩子结点的权值小于右孩子结点的权值），然后分别写出每个字符对应的编码。



对应哈夫曼编码：

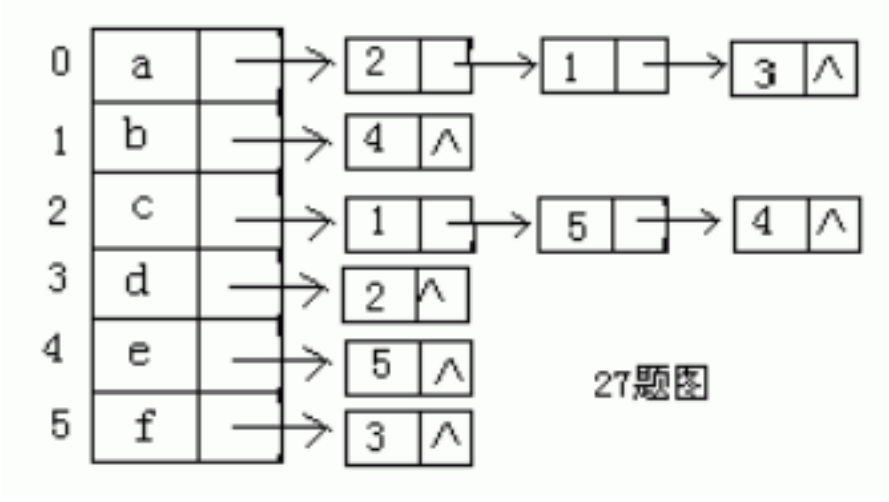
a: 11
b: 1010
c: 100
d: 01
e: 1011
f: 00

27.已知一个图如下所示，其顶点按 a, b, c, d, e, f 顺序存放在邻接表的顶点表中，请画出该图的邻接表，使得按此邻接表进行深度优先遍历时得到的顶点序列为 $acbefd$ ，进行广度优先遍历时得到的顶点序列为 $acbdfe$ 。



题 27 图

答案：



28.已知两个 4×5 的稀疏矩阵的三元组表分别如下：

0	1	4	16	0	1	1	32
1	2	2	18	1	2	2	- 22
2	3	4	- 25	2	2	5	69
3	4	2	28	3	3	4	25
				4	4	2	51

请画出这两个稀疏矩阵之和的三元组表。

解：

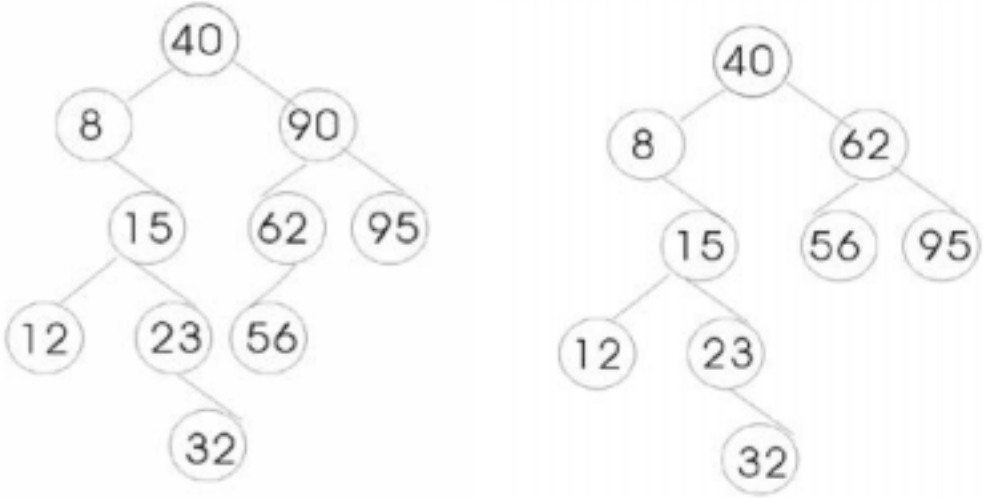
	i	j	w
0	1	1	32
1	1	4	16
2	2	2	-4
3	2	5	69
4	4	2	79

28题图

29.从空树起，依次插入关键字 40，8，90，15，62，95，12，23，56，32，构造一棵二叉排序树。

(1)画出该二叉排序树

(2)画出删去该树中元素值为 90 的结点之后的二叉排序树。



四、算法阅读题（本大题共 4 小题，每小题 5 分，共 20 分）

30.如图所示，利用同一循环向量空间实现两个队列，其类型 Queue2 定义如下：

阅读下列算法，并回答问题：

(1)写出执行函数调用 f(p)的输出结果；

(2)简述函数 f 的功能。

```
void f(BinThrTree t)
{
    while(t)
    {
        printf(t->data);
        if(t->lchild)
            t=t->lchild;
        else
            t=t->rchild;
    }
}
```

答案 (1) ABDFCEGH (2) 先根遍历

32.下列函数 FindCycle(G,i) 的功能是，对一个采用邻接表作存储结构的有向图 G，利用深度优先搜索策略寻找一条经过顶点 v_i 的简单回路。数组 cycle_path 用于保存搜索过程中形成的回路，cycle_path[k]=j(j ≥ 0)表示在回路中顶点 v_k 的下一个顶点是 v_j 。

请在空缺处填入合适的内容，使其成为一个完整的算法。

vertex firstedge

已知邻接表的顶点表结点结构为：

adjvex next

边表结点 EdgeNode 结构为：

int cycle_path[MaxNum];

int FindCycle(ALGraph*G ,int i)

{// 若回路存在，则返回 1，否则返回 0

int j;

for(j=0;j<G->n;j++)cycle_path[j]=-1;

return DFSPath(G,i,i);

}

int DFSPath(ALGraph*G ,int j,int i)

{

EdgeNode *p;

int cycled=0;

for(p=G->adjlist[j].firstedge;p&&!cycled;p=p->next)

{

cycle_path[j]=p->adjvex;

if((1))cycled=1;// 已找到回路

else

if(cycle_path[p->adjvex]==-1)cycled= (2) ;

}

return (3)

} (1) (2) (3)

32 题答案： (1) p->adjvex==i

(2)DFSpath(G,p->adjvex,i)

(3)cycled

33.阅读下列函数 algo,并回答问题。

(1)假设整型数组 A[1..8] 中的元素依次为 (3, 8, 9, 1, 7, 4, 2, 6)。执行函数调用 algo(A,8) 时, 外层 while 的循环体执行多少次?函数的返回值是多少?

(2)简述函数 algo(L,n) 的功能。

```
int algo(int L[],intn)
{
    int i=0,j,s=1,t=n;
    while (i!=(n+1)/2)
    {
        int x=L[s] ;
        i=s;j=t;
        while(i<j)
        {
            while(i<j && L[j]>=x)j--;
            L[i]=L[j];
            while(i<j && L[i]<=x)i++;
            L[j]=L[i];
        }
        L[i]=x;
        if(i<(n+1)/2)s=i+1;
        else t=i-1;
    }
    if(i==0)return 0;
    else return L[i];
} (1) (2) (3)
```

33 题答案:

(1) 外循环执行 4 次, 函数返回值为 3。

(2) 将 A[1] 至 A[8] 中不小于 A[1] 的元素进行递增排序, 如调用 algo(A,8) 时最终排序结果为 2 1 3 4 6 7 8 9

五、算法设计题 (本大题共 10 分)

34.假设以带头结点的单循环链表作非递减有序线性表的存储结构。请设计一个时间复杂度为 $O(n)$ 的算法, 删除表中所有数值相同的多余元素, 并释放结点空间。例如: (7, 10,

10, 21, 30, 42, 42, 42, 51, 70) 经算法操作后变为 (7, 10, 21, 30, 42, 51, 70)

34 题答案:

Exam4(Linklist,L)

```
{listnode *p,*q;
    p=L->next;
    while(p!=L)
    {q=p->next;
        while(q->data==p->data)
        {p->next=q->next;
            free(q);
```

```

        q=p->next;
    }
    p->next;
}
}

```

2003 年 10 月全国数据结构试题 (2006-7-25 2:07:00)

- 计算机识别、存储和加工处理的对象被统称为 (b)
 A. 数据 B. 数据元素
 C. 数据结构 D. 数据类型
- 在具有 n 个结点的有序单链表中插入一个新结点并使链表仍然有序的时间复杂度是 (b)
 A. $O(1)$ B. $O(n)$
 C. $O(n \log n)$ D. $O(n^2)$
- 队和栈的主要区别是 (d)
 A. 逻辑结构不同 B. 存储结构不同
 C. 所包含的运算个数不同 D. 限定插入和删除的位置不同
- 链栈与顺序栈相比，比较明显的优点是 (d)
 A. 插入操作更加方便 B. 删除操作更加方便
 C. 不会出现下溢的情况 D. 不会出现上溢的情况
- 采用两类不同存储结构的字符串可分别简称为 (b)
 A. 主串和子串 B. 顺序串和链串
 C. 目标串和模式串 D. 变量串和常量串
- 在目标串 $T[0..n-1] = \text{xwxxxyxy}$ 中，对模式串 $P[0..m-1] = \text{xy}$ 进行子串定位操作的结果是 (c)
 A. 0 B. 2
 C. 3 D. 5
- 已知广义表的表头为 a ，表尾为 (b,c) ，则此广义表为 (b)
 A. $(a,(b,c))$ B. (a,b,c)
 C. $((a),b,c)$ D. $((a,b,c))$
- 二维数组 A 按行优先顺序存储，其中每个元素占 1 个存储单元。若 $A[1][1]$ 的存储地址为 420， $A[3][3]$ 的存储地址为 446，则 $A[5][5]$ 的存储地址为 (c)
 A. 470 B. 471
 C. 472 D. 473
- 二叉树中第 5 层上的结点个数最多为 (d)
 A. 8 B. 15
 C. 16 D. 32
- 下列编码中属前缀码的是 (a)
 A. $\{1,01,000,001\}$ B. $\{1,01,011,010\}$
 C. $\{0,10,110,11\}$ D. $\{0,1,00,11\}$
- 如果某图的邻接矩阵是对角线元素均为零的上三角矩阵，则此图是 (d)
 A. 有向完全图 B. 连通图

C. 强连通图 D. 有向无环图

12. 对 n 个关键字的序列进行快速排序，平均情况下的空间复杂度为 (d)

A. $O(1)$ B. $O(\log n)$

C. $O(n)$ D. $O(n \log n)$

13. 对表长为 n 的顺序表进行顺序查找，在查找概率相等的情况下，查找成功的平均查找长度为
($n/2$)

A. B.

C. D. n

14. 对于哈希函数 $H(\text{key}) = \text{key} \% 13$ ，被称为同义词的关键字是 (d)

A. 35 和 41 B. 23 和 39

C. 15 和 44 D. 25 和 51

15. 稠密索引是在索引表中 ()

A. 为每个记录建立一个索引项 B. 为每个页块建立一个索引项

C. 为每组记录建立一个索引项 D. 为每个字段建立一个索引项

二、填空题 (每小题 2 分，若有两个空格，每个空格 1 分，共 20 分)

16. 当问题的规模 n 趋向无穷大时，算法执行时间 $T(n)$ 的数量级被称为算法的 (时间复杂度)
_____。

17. 在链表的结点中，数据元素所占的存储量和整个结点所占的存储量之比称作 (存储密度)
_____。

date next

18. 已知链栈的结点结构为 栈顶指针为 top ，则实现将指针 p 所指结点插入栈顶的
语句依次为 _____ 和 _____。

19. 空串的长度是 0 _____；空格串的长度是 (空格的数目) _____。

20. 假设一个 6 阶的下三角矩阵 B 按列优先顺序压缩存储在一维数组 A 中，其中 $A[0]$ 存储矩阵
的第一个元素 b_{11} ，则 $A[14]$ 存储的元素是 b_{63} _____。

21. 在一棵度为 3 的树中，度为 2 的结点个数是 1，度为 0 的结点个数是 6，则度为 3 的结点个数
是 _____ 2 _____。

22. 如图所示的有向无环图可以排出 _____ 种不同的拓扑序列。

23. 利用筛选法将关键字序列 (37, 66, 48, 29, 31, 75) 建成的大根堆为 (_____ 75, 66, 48, 29, 31, 37 _____)。

24. 对长度为 20 的有序表进行二分查找的判定树的高度为 _____ 5 _____。

25. 在多重表文件中，次关键字索引的组织方式是将 _____ 的记录链接成一个链表。

26. 对于单链表、单循环链表和双向链表，如果仅仅知道一个指向链表中某结点的指针 p ，能否
将 p 所指结点的数据元素与其确实存在的直接前驱交换 ? 请对每一种链表作出判断，若可以，写
出程序段；否则说明理由。

date next

单链表和单循环链表的结点结构为

prior date next

双向链表的结点结构为

(1) 单链表 : (不可以，无法找到前驱接点)

(2) 单循环链表 (可以 : $q = p \rightarrow \text{next}; \text{while}(q \rightarrow \text{next} \neq p) q = q \rightarrow \text{next}; q \rightarrow \text{data} \leftrightarrow p \rightarrow \text{data};$

(3) 双向链表 (可以 :p->prior->data<->p->data;)

27. 假设通信电文使用的字符集为 {a,b,c,d,e,f,g} , 字符的哈夫曼编码依次为: 0110, 10, 110, 111, 00, 0111 和 010。

(1) 请根据哈夫曼编码画出此哈夫曼树, 并在叶子结点中标注相应字符;

(2) 若这些字符在电文中出现的频度分别为: 3, 35, 13, 15, 20, 5 和 9, 求该哈夫曼树的带权路径长度。

28. 当采用邻接表作为图的存储结构时, 也可将邻接表中的顶点表由顺序结构改为链表结构。

(1) 请分别画出这种邻接表的顶点链表结点和边表结点, 并说明结点中各个域的作用;

(2) 对如图所示的有向图画出这种邻接表。

29. 已知 4 阶 B- 树如图所示。

(1) 分别画出将关键字 23 和 89 相继插入之后的 B- 树。

(2) 画出从插入之前的 B- 树中删除关键字 51 之后的 B- 树。

四、算法阅读题 (每小题 5 分, 共 20 分)

30. 阅读下列函数 algo, 并回答问题:

(1) 假设队列 q 中的元素为 (2,4,5,7,8), 其中 “ 2 ” 为队头元素。写出执行函数调用 algo(&q) 后的队列 q;

(2) 简述算法 algo 的功能。

```
void algo(Queue *Q)
{
    Stack S;
    InitStack(&S);
    while (!QueueEmpty(Q))
        Push(&S, DeQueue(Q));
    while (! StackEmpty(&S))
        E nQueue(Q,Pop(&S));
}
```

(1)87542

(2) 队列倒置

31. 阅读下列函数 F, 并回答问题:

(1) 已知如图所示的二叉树以二叉链表作存储结构, rt 为指向根结点的指针。写出执行函数调用 F(rt) 的输出结果。

(2) 说明函数 F 的功能。

```
void F(BinTree T)
{
    Stack S;
    if(T)
    {
        InitStack(&S);
        Push(&S,NULL);
        while(T)
        {
```

```

    printf("%c", T->data);
    if(T->rchild) Push(&S,T->rchild);
    if(T->lchild)T=T->lchild;
    else T=Pop(&S);
}
}
}

```

(1)

(2) 前序遍历二叉数

vertex firstedge

32. 已知邻接表的顶点表结点结构为

adjvex next

边表结点 EdgeNode 的结构为

下列算法计算有向图 G 中顶点 v_i 的入度。请在空缺处填入合适的内容，使其成为一个完整的算法。

int FindDegree(ALGraph *G,int i)//ALGraph 为图的邻接表类型

```

{
    int dgree, j;
    EdgeNode *p;
    degree= (1) ;
    for(j=0;j<G->n;j++)
    {
        p=G->adjlist [ j ] . firstedge;
        while ( (2) )
        {
            if( (3) )
            {
                degree++;
                break;
            }
            p=p->next;
        }
    }
    return degree;
}

```

(1)degree=0;

(2)p

(3)p->adj==vi

data next

33. 已知单链表的结点结构为

下列算法对带头结点的单链表 L 进行简单选择排序，使得 L 中的元素按值从小到大排列。
请在空缺处填入合适的内容，使其成为完整的算法。

```

void SelectSort(LinkedList L)
{
    LinkedList p,q,min;
    DataType rcd;
    p= (1) ;
    while(p!=NULL) {
        min=p;
        q=p->next;
        while(q!=NULL){
            if( (2) )min=q;
            q=q->next;
        }
        if( (3) ){
            rcd=p->data;
            p->data=min->data;
            min->data=rcd;
        }
        (4) ;
    }
}

```

(1)L->next

(2)q->data<min->data

(3)p!=min

(4)p=p->next

五、算法设计题 （ 本题 10 分）

34. 设线性表 $A=(a_1,a_2,a_3,,a_n)$ 以带头结点的单链表作为存储结构。编写一个函数，对 A 进行调整，使得当 n 为奇数时 $A=(a_2,a_4,,,a_{n-1},a_1,a_3,,,a_n)$ ，当 n 为偶数时 $A=(a_2,a_4,,,a_n,a_1,a_3,,,a_{n-1})$ 。

```

typedef struct node

```

```

{
    int x;
    struct node *next;
} NODE;

```

```

typedef NODE * LinkList;

```

```

void adjust( LinkList header )

```

```

{
    NODE *pTmp = header; //      用来保存偶数链表尾指针
    NODE *pCur = header->next; //      链表遍历指针
    NODE *pOddHdr = header->next;//      奇数链表头指针
    NODE *pOddTail = header->next;//      奇数链表尾指针
    int blsOdd = true; //      奇数结点标志，第一个结点是奇数结点

```

```

if( NULL == pCur )//      空链表，不需要处理
return;

while( pCur->next != NULL )//      从第二个结点开始遍历
{
    pCur = pCur->next;
    blsOdd = !blsOdd;
    if( blsOdd )          //      （这步错误，未将原链表的接点连接）
    { //      奇数结点，加入奇数链表表尾
        pOddTail->next = pCur;
        pOddTail = pCur;
    }
    else
    { //      偶数结点，加入偶数链表表尾
        pTmp->next = pCur;
        pTmp = pCur;
    }
}
pOddTail->next = NULL;//      奇数链表表尾结点的 next 置空
pTmp->next = pOddHdr;//      奇数链表插入偶数链表表尾 （这步错误，未考虑最后接点的奇偶性。）

return;
}

```

全国 2004 年 1 月高等教育自学考试

数据结构试题

课程代码： 02331

一、单项选择题（本大题共 15 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，请将其代码填写在题后的括号内。错选、多选或未选均无分。

- 1．在数据结构中，数据的逻辑结构可以分成（ ）
 - A．内部结构和外部结构
 - B．线性结构和非线性结构
 - C．紧凑结构和非紧凑结构
 - D．动态结构和静态结构
- 2．在以单链表为存储结构的线性表中，数据元素之间的逻辑关系用（ ）
 - A．数据元素的相邻地址表示
 - B．数据元素在表中的序号表示
 - C．指向后继元素的指针表示
 - D．数据元素的值表示
- 3．设 p 指向单链表中的一个结点，s 指向待插入的结点，则下述程序段的功能是（ ）


```

s -> next = p -> next; p -> next = s;

t = p -> data; p -> data = s -> data; s ->data = t;

```

 - A．结点 *p 与结点 *s 的数据域互换

- B. 在 p 所指结点的元素之前插入元素
C. 在 p 所指结点的元素之后插入元素
D. 在结点 *p 之前插入结点 *s

4. 栈和队列都是 ()

- A. 限制存取位置的线性结构 B. 顺序存储的线性结构
C. 链式存储的线性结构 D. 限制存取位置的非线性结构

5. 若数组 $s[0..n-1]$ 为两个栈 s1 和 s2 的共用存储空间, 且仅当 $s[0..n-1]$ 全满时, 各栈才不能进行进栈操作, 则为这两个栈分配空间的最佳方案是: s1 和 s2 的栈顶指针的初值分别为 ()

- A. 1 和 $n+1$ B. 1 和 $n/2$
C. -1 和 n D. -1 和 $n+1$

6. 执行下列程序段后, 串 X 的值为 ()

```
S= " abcdefgh " ; T= " xyzw " ;  
substr (X,S,2,strlen(T));  
substr (Y,S, strlen(T),2);  
strcat (X,Y);
```

- A. " cdefgh " B. " cdxyzw "
C. " cdefxy " D. " cdefef "

7. 多维数组之所以有行优先顺序和列优先顺序两种存储方式是因为 ()

- A. 数组的元素处在行和列两个关系中 B. 数组的元素必须从左到右顺序排列
C. 数组的元素之间存在次序关系 D. 数组是多维结构, 内存是一维结构

8. 从广义表 $LS = ((p, q), r, s)$ 中分解出原子 q 的运算是 ()

- A. tail (head (LS)) B. head (tail (head (LS)))
C. head (tail (LS)) D. tail (tail (head (LS)))

9. 在具有 n 个叶子结点的严格二叉树中, 结点总数为 ()

- A. $2n+1$ B. $2n$
C. $2n-1$ D. $2n-2$

10. 若 $\langle v_i, v_j \rangle$ 是有向图的一条边, 则称 ()

- A. v_i 邻接于 v_j B. v_j 邻接于 v_i
C. v_i 和 v_j 相互邻接 D. v_i 与 v_j 不相邻接

11. 在一个带权连通图 G 中, 权值最小的边一定包含在 G 的 ()

- A. 最小生成树中 B. 深度优先生成树中
C. 广度优先生成树中 D. 深度优先生成森林中

12. 当在二叉排序树中插入一个新结点时, 若树中不存在与待插入结点的关键字相同的结点, 且新结点的关键字小于根结点的关键字, 则新结点将成为 ()

- A. 左子树的叶子结点 B. 左子树的分支结点
C. 右子树的叶子结点 D. 右子树的分支结点

13. 希尔排序的增量序列必须是 ()

- A. 递增的 B. 随机的
C. 递减的 D. 非递减的

14. 如果在排序过程中, 每次均将一个待排序的记录按关键字大小加入到前面已经有序的子表中的适当位置, 则该排序方法称为 ()

- A. 插入排序 B. 归并排序

C. 冒泡排序 D. 堆排序

15. 设置溢出区的文件是 ()

A. 索引非顺序文件 B. ISAM 文件

C. VSAM 文件 D. 顺序文件

二、填空题 (本大题共 10 小题, 每小题 2 分, 共 20 分)

请在每小题的空格中填上正确答案。错填、不填均无分。

16. 下列程序段的时间复杂度为 $O(n^2)$

```
product = 1;
for (i = n; i > 0; i--)
    for (j = i + 1; j < n; j++)
        product *= j;
```

17. 已知指针 p 指向单链表中某个结点, 则语句 `p->next = p->next->next` 的作用是 _____。

删除 *P 的直接后继结点


18. 假设元素只能按 a,b,c,d 的顺序依次进栈, 且得到的出栈序列中的第一个元素为 c, 则可能得到的出栈序列为 _____, 不可能得到的出栈序列为 _____

1)cbad, cbda, cdba

2)cabd, cadb, cdab

19. 若链串结点中的指针占 4 个字节, 每个字符占 1 个字节, 则结点大小为 2 的链串的存储密度为 _____。

$2 / (4 + 2) = 1/3$

20. 右图表示的广义表为 _____。 [img]/  <http://www.ezikao.com.cn/uploadimages/20046615335847449.jpg> [img]

((e), ((e), (b, c)), (L))

21. 若一棵满三叉树中含有 121 个结点, 则该树的深度为 _____。

$5 \quad // (3^5 - 1) / (3 - 1) = 121$

22. 若以邻接矩阵表示有向图, 则邻接矩阵上

第 i 行中非零元素的个数即为顶点 v_i 的 _____。
出度

23. 若希望只进行 8 趟排序便能在 4800 个元素中找出其中值最小的 8 个元素, 并且要求排序过程中所进行的关键字比较次数尽可能少, 则应该选用 _____ 排序方法。

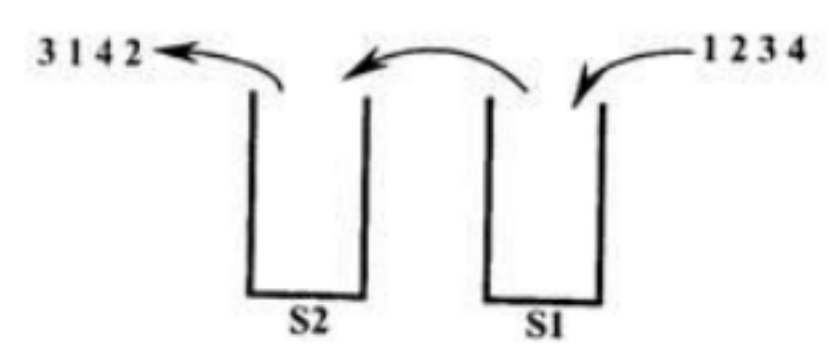
24. 在含 20 个关键字的 3 阶 B 树 (2 - 3 树) 上查找一个关键字, 至多需要访问 _____ 次外存。

25. 文件上的两类主要操作为 _____ 和 _____。

检索 和 维护

三、解答题（本大题共 4 小题，每小题 5 分，共 20 分）

26 . 设栈 S1 的入栈序列为 1 2 3 4 （每个数字为 13 个元素），则不可能得到出栈序列 3142。但可通过增设栈 S2 来实现。例如，按下图中的箭头指示，依次经过栈 S1 和 S2，便可得到序列 3 1 4 2 。



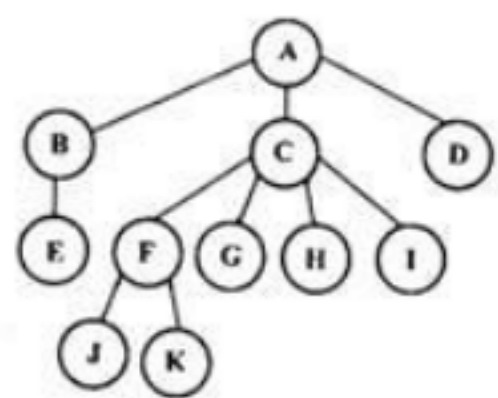
题 26 图

如果用 H1 和 H2 分别表示栈 S1 和 S2 的进栈操作，用 P1 和 P2 分别表示两个栈的出栈操作，则得到 3 1 4 2 的一个操作步骤为

H1 , H1 , H1 , P1 , H2 , P2 , P1 , H2 , P1 , H2 , P2 , H1 , P1 , H2 , P2 , P2

请仿照上例写出利用两个栈从 1 2 3 4 得到 4 1 3 2 的操作步骤。

H1,P1,H2,H1,H1,H1,P1,H2,P2,P2,P1,H2,P2,P1,H2,P2



题 27 图

27 . 已知树如右图所示，

(1) 写出该树的后序序列；

(2) 画出由该树转换得到的二叉树。

1) EBJKFGHICDA

2) 树变二叉树：兄弟相连，保留长子的连线

. A
. /
. B
. / \\
. E C
. / \\
. F D
. /\n
. J G
. \ \n
. K H
. \n
. I

28. 为关键字 (17, 33, 31, 40, 48) 构造一个长度为 7 的散列表, 设散列函数为 $h(key)=key\%7$, 用开放定址法解决冲突的探查序列是

$$h_i = (h(key) + i(key\%5+1))\%7 \quad 0 \leq i \leq 6$$

(1) 画出构造所得的散列表;

(2) 求出在等概率情况下查找成功时的平均查找长度。

1)

. 0 1 2 3 4 5 6

. 31 17 48 33 40

2) $(1 + 1 + 3 + 2 + 4) / 5 = 11 / 5$

29. 已知 $R[1..8]$ 中的元素依次为 (12, 5, 9, 20, 6, 31, 24, 27), 写出按算法 MergeSortDC 对 R 进行自顶向下的二路归并排序过程中, 前 5 次调用函数 Merge(R, low, mid, high) 时参数 low, mid 和 high 的值。

```
void MergeSortDC (int R[], int low, int high )
```

```
{
    int mid
    if (low<high)
    {
        mid = (low +high)/2;
        MergeSortDC (R, low, mid);
        MergeSortDC (R, mid+1, high);
        Merge (R, low, mid, high);
    }
}
```

```
// MergeSortDC
```

(1) 第一次调用时的参数值;

(2) 第二次调用时的参数值;

(3) 第三次调用时的参数值;

(4) 第四次调用时的参数值;

(5) 第五次调用时的参数值;

// 此题有一定的难度, 涉及到递归调用时, 系统堆栈的情况

. low mid high

1) 1 1 2

2) 3 3 4

3) 1 2 4

4) 5 5 6

5) 7 7 8

6) 5 8 6

7) 1 8 4

四、算法阅读题 (本大题共 4 小题, 每小题 5 分, 共 20 分)

30. 下列函数的功能是, 对以带头结点的单链表作为存储结构的两个递增有序表 (表中不存在值相同的数据元素) 进行如下操作: 将所有在 Lb 表中存在而 La 表中不存在的结点插入到 La 中, 其中 La 和 Lb 分别为两个链表的头指针。请在空缺处填入合适内容, 使其成为一个完整的算法。

```
void union (LinkList La, LinkList Lb)
```



```

{
// 本算法的功能是将所有 Lb 表中存在而 La 表中不存在的结点插入到 La 表中
LinkedList pre = La, q;
LinkedList pa = La -> next;
LinkedList pb = Lb -> next;
free (Lb);
while (pa && pd)
{
if (pa -> data < pb -> data)
{ pre = pa; pa = pa -> next;}
else if (pa -> data > pb -> data)
{
(1) ;
pre = pb;
pb = pb -> next;
(2) ;
}
else
{
q = pb; pb = pb -> next; free (q);
}
}
if (pb)
(3) ;
}

```

(1) pre->next = pb

(2) pre->next = pa

(3) pre->next = pb

31 . 已知串的存储结构为动态存储分配的顺序串。阅读下列算法，并回答问题：

(1) 写出执行函数调用 strc (s, r) 的返回结果，其中 s= " aba " , r= " abababa " ；

(2) 简述函数 strc 的功能。

```

int strc (HString * sub, HString * str)
{
int i=0, j, k, count =0;
while (i < str -> length - sub -> length +1)
{
j=i; k=0;
while (k <sub -> length && str -> ch[j] ==sub -> ch[k] )
{
j++; k++;
}
if (k == sub -> length)

```

```

{count++; i=j-sub -> length +1;}
else i++;
}
return count;
}

```

(1) 3

(2) 求串 str 中子串 sub 的个数

32 . 下列函数 MDFSTree 的功能是 , 对一个采用邻接矩阵作存储结构的图进行深度优先搜索遍历 , 输出所得深度优先生成森林中各条边。 请在空缺处填入合适内容 , 使其成为一个完整的算法。

```

#define MaxMun 20 // 图的最大顶点数
typedef struct {
    char vexs [MaxNum]; // 字符类型的顶点表
    int edges [MaxNum][MaxNum]; // 邻接矩阵
    int n, e; // 图的顶点数和边数
}MGraph; // 图的邻接矩阵结构描述

typedef enum {FALSE, TRUE} Boolean;
Boolean visited [MaxNum];

void MDFSTree (MGraph *G, int i);
void MDFSTree (MGraph *G)
{
    int i, k;
    for (i=0; i <G -> n; i++)
        visited [i] = (1) ;
    for (k = 0; k<G -> n; k++)
        if (!visited [k]) MDFSTree (G,k);
}

void MDFSTree (MGraph *G, int i)
{
    int j;
    visited [i]=TRUE;
    for (j=0; j<G -> n; j++)
    {
        if ( (2) )
        {
            printf ( " <%c, %c> " G -> vexs [i], G -> vexs [j]);
            (3) ;
        }
    }
}

1) FALSE // 初始化 , 所有结点未访问

```

2) !visited [j] && G->edge[i][j] == 1 // 结点 j 未访问且 i 到 j 有边
 3) MDFSTree(G, j) // 从 j 结点继续，深度优先搜索

33. 已知整形数组 L[1..8] 中的元素依次为 (9 , 8 , 5 , 7 , 6 , 3 , 2 , 1) , 阅读下列函数 , 并写出执行函数调用 sort(L, 8) 时 , 对 L 进行的头两趟 (pass 分别为 0 和 1) 处理结果。

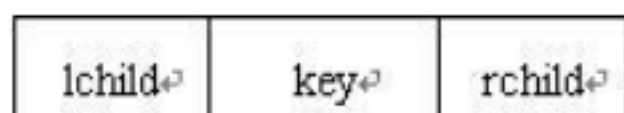
```
Void sort (int R[],int n)
{
    int pass = 0, k, exchange, x;
    do {
        k=pass%2+1;
        exchange = 0;
        while (k<n)
        {
            if (R[k]>R[k+1])
            {
                x = R[k]; R[k] = R[k+1]; R[k+1] = x;
                exchange =1;
            }
            K+=2
        }
        pass ++;
    }while (exchange == 1|| pass <=1);
}
```

第一趟 (pass = 0) : 8 9 5 7 3 6 1 2

第二趟 (pass = 1) : 8 5 9 3 7 1 6 2

五、算法设计题 (本大题共 10 分)

34 . 已知二叉排序树中结点的关键字为整数 , 设计递归算法按递增有序性输出树中所有大于或等于给定值 x 的结点 , 并以函数的参数返回输出的结点个数。 假设以二叉链表为存储结构 , 其结点结构为 :



```
void find( BT * root, int x, int * count )
{
    if( !root ) return;
    if( root->key >= x ){// 因为是排序树 , 只有当 key>=x 时 , 才需要查找其左子树
        if( root->lchild ) find( root->lchild, x , count );
        (*count)++;
        printf("%d\t", root->key );
    }
    if( root->rchild ) find( root->rchild, x, count );
}
```

全国 2004 年 10 月卷答案

一、单项选择题

DABAC CCBDA ABABD

// 5. 可以简单的计算，空域为 3->7，总共 5 个，对长则为 21 - 5 = 16

7.c//BDBABDABDAB

BDA

123 失败，比较 3 次

BDBABDABDAB

BDA

1 失败，比较 1 次

BDBABDABDAB

BDA

12 失败，比较 2 次

BDBABDABDAB

BDA

1 失败，比较 1 次

BDBABDABDAB

BDA

123 成功，比较 3 次 共计 10 次

10.d

A
/
B
/ | \

C D F
|
E

二、填空题

16. (一组) 运算

17. 直接前驱

18. SXSSXXSSXSSXXX

19. 模式匹配

20. 5n - 6

$N+2N-2+2N-4=5N-6$

// n 阶 5 对角阵

// 1 1 1 0 0

// 1 1 1 1 0

// 1 1 1 1 1 0

// 0 1 1 1 1 1 0

// 0 0 1 1 1 1 1 0

//0 1 1 1 1 1 0

//

//

21. 50

// $63 < 100 < 127$, 最下一层叶子数 : $100 - 63 = 37$

// 倒数第 2 层叶子数 : $32 - \lceil 37 / 2 \rceil = 13$ [] 向上取整

22. 径 ?

23. 待排关键字 (记录) ?

24. 有序的 ?

25. ?

// 一些概念题 , 因为没书 , 很久没接触了 , 可能不准确。

三、解答题

略

28 划分后左边 : (55) (28) (73) (91) (37) 右边 : (64), (19), (82), (46)

第一次 Merge 之后 : (28,55)(73) (91) (37) 右边 : (64), (19), (82), (46)

第二次 Merge 之后 : (28,55)(73,91) (37) 右边 : (64), (19), (82), (46)

第三次 Merge 之后 : (28,55,73,91)(37) 右边 : (64), (19), (82), (46)

第四次 Merge 之后 : (28,37,55,73,91) 右边 : (64), (19), (82), (46)

第五次 Merge 之后 : (28,37,55,73,91) 右边 : (19,64), (82), (46)

所以28 , 37 , 55 , 73 , 91 , 19 , 64 , 82 , 46

四、算法阅读题

30.

1) $p = pre \rightarrow next$; 或 $p = L \rightarrow next$; // p 指向第一个结点

2) $p \rightarrow next = Lc \rightarrow next$; // 数据大于 c 的 p 结点插入 Lc 链表表头

3) $p = pre \rightarrow next$; 或 $p = p \rightarrow next$; // 下一个结点

31. 此题有误 , ... if ((i!=t)!=0) ... 应该是 ... if ((i = !i) != 0) ...

1) 1,3,5,7,6,4,2

2) 堆栈 S 中的元素依次出栈 , 奇数次序的入栈 T, 偶数次序的入队 Q

32. 图 G 的邻接矩阵不对称 , 因此 , 是有向图

1) 5

2) 计算有向图 G 中的端点 i (第 i+1 个端点) 的度, 包括出度和入度
3) O(n)

33. 此题明显有错误 if(low > high) 应为 if(low < high)

因为 if(){...} 里有 while(low < high)

1) -8, -3, -2, -1, 4, 2, 5, 7

: -8 -3 2 -1 -2 4 5 7

2) 将数组 R 中的前 n 个数调整为所有负数在前, 所有整数在后

五、算法设计题

34. 看原型, 应该是要使用递归了, 题目很傻地把解法都告诉我们的。

f34(BinTree T,int level,int *lmin,int *lmax)

```
{
    if(T){
        if( T->lchild == NULL && T->rchild == NULL ){
            if( *lmin == 0 || level < *lmin ) *lmin = level;
            if( level > *lmax ) *lmax = level;
            return;
        }
        if( T->lchild )
            f34( T->lchild, level + 1, lmin, lmax );
        if( T->rchild )
            f34( T->rchild, level + 1, lmin, lmax );
    }
}
```

2005.1 全国卷答案 (18 号更新)

一、单项选择题 BDBBB ADDCA CBACC

二、填空题

16. O(n)

17. p->next && p->next->next == NULL

18. 41

19. 0?

20. $1100 + 2 * (4 * 6 * 7 + 3 * 7 + 2) = 1482$

21. CBDA

22. n - 1

23. 3 // 56 前面, 比 56 大的数的个数 + 1

24. 表结点的个数

25. $\lg n$

三、解答题

26.

1) $((a), ((b), c))$

27.

[方法 1, liangliangzai]

b 个非叶子节点, 有 $k \cdot b$ 个孩子,

加上根节点, 节点总数: $k \cdot b + 1$

节点总数 = 非叶节点 + 叶节点 = $a + b$

得 $a = (k - 1)b + 1$

[方法 2]

设满 k 叉树的高度为 n , 则:

叶子结点数 $a = k^{(n-1)}$

非叶结点数 $b = 1 + k + k^2 \dots + k^{(n-2)} = [k^{(n-1)} - 1] / (k - 1)$

$\Rightarrow b = (a - 1) / (k - 1) \Rightarrow a = (k - 1)b + 1$

28

最短路径长度	已确定点集	最短路径直接前趋
b c d e f	b c d e f	
20 60 * 10 65 (a)	a a * a a	
20 60 * - 30 (a e)	a a * a e	
- 50 * - 30 (a e b)	a b * a e	
- 45 110 - - (a e b f)	a f f a e	
- - 85 - - (a e b f c)	a f c a e	
- - - - - (a e b f c d)	a f c a e	

所以 $a \rightarrow b$ 20 a,b

$a \rightarrow e$ 10 a,e

$a \rightarrow f$ 30 a,e,f

$a \rightarrow c$ 45 a,e,f,c

$a \rightarrow d$ 85 a,e,f,c,d

29

48 70 33 92 24 56 12 65

48 70 56 92 24 33 12 65

48 92 56 70 24 33 12 65

92 70 56 65 24 33 12 48

四、算法阅读题

30.1) $Q \rightarrow rear == Q \rightarrow front \ \&\& \ tag == 1;$

2) $\text{if}(Q \rightarrow rear == Q \rightarrow front) \ tag = 1;$

3) $Q \rightarrow rear == Q \rightarrow front \ \&\& \ tag == 0;$

4) $Q \rightarrow \text{front} == (Q \rightarrow \text{front} + 1) \% \text{MAXQSIZE};$

5) $\text{if}(Q \rightarrow \text{rear} == Q \rightarrow \text{front}) \text{tag} = 0;$

31.1)ABDECF

2) 先根遍历 BT

32.1)3,3,3,1,1,1,2,2,2 swap() 调用 6 次

2) 将 $f[]$ 中的元素，所有值为 x 的放在 y 前，其他的放在所有 y 后。

33.1) $(\text{low} + \text{high}) / 2$

2) $\text{low} = \text{mid} + 1$

// while 循环结束时， $\text{low} = \text{high} + 1$

3) $r \geq \text{low}$

4) $r[\text{low}] = x$

五、算法设计题

34.

`void f35(LinkList h, LinkList *h1, LinkList *h2)`

`{`

`LinkList pre = h, p = h->next, ph2;`

`*h1 = h;`

`*h2 = malloc(sizeof(struct PNode));`

`*h2->next = *h2;`

`ph2 = *h2;`

`while(p != h){`

`if((p->exp) % 2){// 奇次项保留`

`pre = p;`

`p = pre->next;`

`}`

`else{// 偶次项插入 *h2`

`pre->next = p->next;// 偶次项脱离 h`

`ph2->next = p; // 尾插入 *h2`

`p->next = *h2;`

`ph2 = p;`

`p = pre->next;`

`}`

`}`

`}`

全国 2005 年 10 月高等教育自学考试

数据结构试题

课程代码： 02331

一、单项选择题（本大题共 15 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，请将其代码填写在题后的括号内。错选、多选或未选均无分。

1. 若将数据结构形式定义为二元组 (K, R) ，其中 K 是数据元素的有限集合，则 R 是 K 上 ()
A. 操作的有限集合 B. 映象的有限集合
C. 类型的有限集合 D. 关系的有限集合
2. 在长度为 n 的顺序表中删除第 i 个元素 ($1 \leq i \leq n$) 时，元素移动的次数为 ()
A. $n-i+1$ B. i
C. $i+1$ D. $n-i$
3. 若不带头结点的单链表的头指针为 $head$ ，则该链表为空的判定条件是 ()
A. $head == NULL$ B. $head \rightarrow next == NULL$
C. $head != NULL$ D. $head \rightarrow next == head$
4. 引起循环队列队头位置发生变化的操作是 ()
A. 出队 B. 入队
C. 取队头元素 D. 取队尾元素
5. 若进栈序列为 $1, 2, 3, 4, 5, 6$ ，且进栈和出栈可以穿插进行，则不可能出现的出栈序列是 ()
A. $2, 4, 3, 1, 5, 6$ B. $3, 2, 4, 1, 6, 5$
C. $4, 3, 2, 1, 5, 6$ D. $2, 3, 5, 1, 6, 4$
6. 字符串通常采用的两种存储方式是 ()
A. 散列存储和索引存储 B. 索引存储和链式存储
C. 顺序存储和链式存储 D. 散列存储和顺序存储
7. 设主串长为 n ，模式串长为 m ($m \leq n$)，则在匹配失败情况下，朴素匹配算法进行的无效位移次数为 ()
A. m B. $n-m$
C. $n-m+1$ D. n
8. 二维数组 $A[12][18]$ 采用列优先的存储方法，若每个元素各占 3 个存储单元，且第 1 个元素的地址为 150，则元素 $A[9][7]$ 的地址为 ()
A. 429 B. 432
C. 435 D. 438
9. 对广义表 $L = ((a,b),(c,d),(e,f))$ 执行操作 $tail(tail(L))$ 的结果是 ()
A. (e,f) B. $((e,f))$
C. (f) D. $()$
10. 下列图示的顺序存储结构表示的二叉树是 (A)

18. 栈下溢是指在 ____ 栈空 ____ 时进行出栈操作。
19. 已知 `substr(s,i,len)` 函数的功能是返回串 `s` 中第 `i` 个字符开始长度为 `len` 的子串，`strlen(s)` 函数的功能是返回串 `s` 的长度。若 `s="ABCDEFGHIIJK,t="ABCD`，执行运算 `substr(s,strlen(t), strlen(t))` 后的返回值为 ____ "DEFG" // 注意双引号不能少 ____。
20. 去除广义表 `LS=(a1,a2,, , an)` 中第 1 个元素，由其余元素构成的广义表称为 `LS` 的 __表尾__。
21. 已知完全二叉树 `T` 的第 5 层只有 7 个结点，则该树共有 ____ $2^3 + 7 / 2 = 11$ ____ 个叶子结点。
22. 在有向图中，以顶点 `v` 为终点的边的数目称为 `v` 的 ____入度____。
23. 当关键字的取值范围是实数集合时，无法进行箱排序和 ____基数____排序。
24. 产生冲突现象的两个关键字称为该散列函数的 ____同义词____。
25. 假设散列文件中一个桶能存放 `m` 个记录，则桶“溢出”的含义是，当需要插入新的记录时，该桶中 ____已有 `m` 个同义词记录 ____。

三、解答题（本大题共 4 小题，每小题 5 分，共 20 分）

26. 假设以数组 `seqn [m]` 存放循环队列的元素，设变量 `rear` 和 `quelen` 分别指示循环队列中队尾元素的位置和元素的个数。

- (1) 写出队满的条件表达式；
- (2) 写出队空的条件表达式；
- (3) 设 `m=40, rear=13, quelen=19`，求队头元素的位置；
- (4) 写出一般情况下队头元素位置的表达式。

- (1) `quelen == m`
- (2) `quelen == 0`
- (3) `(13 - 19 + 40) % 40 = 34`
- (4) `(rear - quelen + m) % m`

27. 已知一棵二叉树的中序序列为 `ABCDEFQ` 层序序列为 `BAFEGCD` 请画出该二叉树。

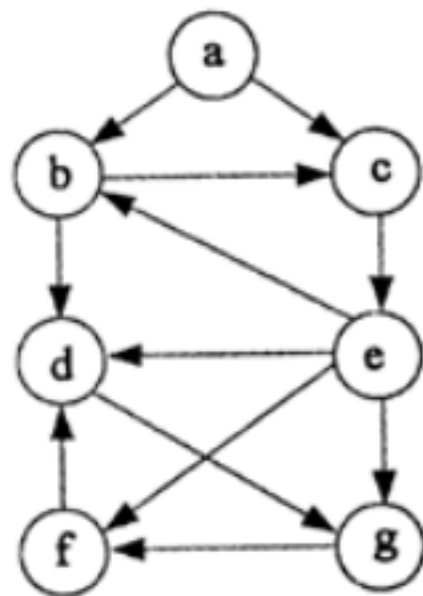
- 1) (A), B ,(CDEFG)
- 2) (A), B ,((CDE), F, (G))

.....

```

      B
     / \
    A   F
     /\
    E   G
     /
    C
     \
    D
  
```

28. 画出下图所示有向图的所有强连通分量。



题 28 图

3 个: a 、 bce 、 dfg

29. 对 7 个关键字进行快速排序，在最好的情况下仅需进行 10 次关键字的比较。

(1) 假设关键字集合为 {1,2,3,4,5,6,7}，试举出能达到上述结果的初始关键字序列；

(2) 对所举序列进行快速排序，写出排序过程。

我们知道，对 n 个关键字序列进行一趟快速排序，要进行 n-1 次比较，

也就是基准和其他 n-1 个关键字比较。

这里要求 10 次，而 $7 - 1 + 2 * (3 - 1) = 10$ ，这就要求 2 趟快速排序后，算法结束。

所以，列举出来的序列，要求在做 partition 的时候，正好将序列平分

(1) 4 1 3 2 6 5 7

或 4 1 3 7 6 5 2

或 4 5 3 7 6 1 2

或 4 1 3 5 6 2 7

(2) 自己列吧 :)

四、算法阅读题（本大题共 4 小题，每小题 5 分，共 20 分）

30. 阅读下列算法，并回答问题：

(1) 设顺序表 L=(3,7,11,14,20,51)，写出执行 f30(&L,15) 之后的 L；

(2) 设顺序表 L=(4,7,10,14,20,51)，写出执行 f30(&L,10) 之后的 L；

(3) 简述算法的功能。

```

void f30(SeqList*L, DataType x)
{
    int i=0, j;
    while (i<L->length && x>L->data[i]) i++;
    if(i<L->length && x==L->data[i]) { // 找到 x，则删除 x，大于 x 的数前移
        for(j=i+1; j<L->length; j++)
            L->data[j-1] = L->data[j];
        L->length--;
    } else { // 没找到，插入 x，大于 x 的数后移
        for(j=L->length; j>i; j--)
            L->data[j] = L->data[j-1];
        L->data[i] = x;
        L->length++;
    }
}
  
```

```

    }
}
(1) L=(3,7,11,14,15,20,51)
(2) L=(4,7,14,20,51)
(3) 在顺序表 L 中查找数 x,
    找到, 则删除 x,
    没找到, 则在适当的位置插入 x, 插入后, L 依然有序.

```

31. 已知图的邻接表表示的形式说明如下：

```

#define MaxNum 50      //      图的最大顶点数
typedef struct node {
    int  adjvex;        //      邻接点域
    struct node *next;  //      链指针域
} EdgeNode;           //      边表结点结构描述
typedef struct {
    char vertex;        //      顶点域
    EdgeNode *firstedge; //      边表头指针
} VertexNode; //      顶点表结点结构描述
typedef struct {
    VertexNode adjlist [ MaxNum] ; //      邻接表
    int n, e;           //      图中当前的顶点数和边数
} ALGraph; //      邻接表结构描述

```

下列算法输出图 G 的深度优先生成树（或森林）的边。阅读算法，并在空缺处填入合适的内容，使其成为一个完整的算法。

```

typedef enum {FALSE, TRUE} Boolean;
Boolean visited [ MaxNum] ;
void DFSForest(ALGraph *G){
    int i;
    for(i=0;i<G->n;i++) visited [ i ] = _____(1);
    for(i=0;i<G->n;i++) if (!visited [ i ] ) DFSTree(G,i);
}
void DFSTree(ALGraph *G, int i) {
    EdgeNode *p;
    visited [ i ] =TRUE;
    p=G->adjlist [ i ] . firstedge;
    while(p!=NULL){
        if(!visited [ p->adjvex ] ){
            printf(    <%c,%c> ,G ->adjlist [ i ] . vertex,
                    G->adjlist [ p->adjvex ] . vertex);
            _____(2) ;
        }
        _____(3) ;
    }
}

```

(1) FALSE // 初始化为未访问

(2) DSFTree(G, p->adjvex); // 从相邻结点往下继续深度搜索

(3) p = p->next; // 下一个未访问的相邻结点

32. 阅读下列算法，并回答问题：

(1) 假设数组 $L[8] = \{3, 0, 5, 1, 6, 4, 2, 7\}$ ，写出执行函数调用 $f32(L, 8)$ 后的 L ；

(2) 写出上述函数调用过程中进行元素交换操作的总次数。

```
void f32(int R[ ], int n){
    int i, t;
    for (i=0; i<n-1; i++)
        while (R[i] != i){
            t=R[ R[i] ];
            R[ R[i] ] = R[i];
            R[i] = t;
        }
}
```

while(){} 里是把 $R[i]$ 和 $R[R[i]]$ 交换；

(1) $L = \{0, 1, 2, 3, 4, 5, 6, 7\}$;

(2) 5 次

33. 已知带头结点的单链表中的关键字为整数，为提高查找效率，需将它改建为采用拉链法处理冲突的散列表。设散列表的长度为 m ，散列函数为 $\text{Hash}(\text{key}) = \text{key} \% m$ 。链表的结点结构为：。请在空缺处填入适当内容，使其成为一个完整算法。

```
void f33 (LinkedList L, LinkedList H[ ], int m)
{// 由带头结点的单链表 L 生成散列表 H，散列表生成之后原链表不再存在
    int i, j;
    LinkedList p, q;
    for (i=0; i<m; i++)
        H[i] = _____(1);
    p=L->next;
    while(p)
    {
        q=p->next;
        j=p->key%m;
        _____(2);
        H[j] = p;
        _____(3);
    }
    free(L);
}
```

(1) NULL // 初始化

(2) $p \rightarrow \text{next} = H[j]$ // 和下面一句完成头插法

(3) $p = q$; // 继续遍历 L

五、算法设计题（本大题 10 分）

34. 假设以带双亲指针的二叉链表作为二叉树的存储结构，其结点结构的类型说明如下所示：

```

typedef char DataType;
typedef struct node {
    DataType data;
    struct node *lchild, *rchild;    //          左右孩子指针
    struct node *parent;            //          指向双亲的指针
} BinTNode;

```

```

typedef BinTNode *BinTree;

```

若 px 为指向非空二叉树中某个结点的指针，可借助该结构求得 px 所指结点在二叉树的中序序列中的后继。

(1) 就后继的不同情况，简要叙述实现求后继操作的方法；

(2) 编写算法求 px 所指结点的中序序列后继，并在算法语句中加注注释。

1)

a)*px 有右孩子，则其右孩子为其中序序列中的后继

b)*px 无右孩子，从 *px 开始回溯其祖先结点，找到第 1 个身份为左孩子的结点，找到，则该结点的父结点为 *px 的中序序列中的后继。找不到，则无后继。

2)

```

BinTNode * fintNext( BinTNode * px )

```

```

{
    if( px-> rchild ) return px->rchild; /*px          有右孩子
    BinTNode *q, *qp;
    q = px;
    while( qp = q->parent ){ //          未回溯到根结点
        if( qp->lchild == q ) return qp; //          找到 1)b) 所述结点
        q = qp; //          往上回溯
    }
    return NULL; //          未找到
}

```

全国 2006 年 1 月高等教育自学考试

数据结构试题

课程代码： 02331

一、单项选择题（本大题共 15 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，请将其代码填写在题后的括号内。错选、多选或未选均无分。

1．根据数据元素的关键字直接计算出该元素存储地址的存储方法是（ D ）

A．顺序存储方法 B．链式存储方法

C．索引存储方法 D．散列存储方法

2．下述程序段中语句 的频度是（ C ）

```

s=0;
for(i=1;i<m;i++)
for(j=0;j<=i;j++)
    s+=j;

```

A． B．

C． D．

3. 求单链表中当前结点的后继和前驱的时间复杂度分别是 (C)
A. $O(n)$ 和 $O(1)$ B. $O(1)$ 和 $O(1)$
C. $O(1)$ 和 $O(n)$ D. $O(n)$ 和 $O(n)$
4. 非空的单循环链表的头指针为 head,尾指针为 rear,则下列条件成立的是 (A)
A. $rear \rightarrow next == head$ B. $rear \rightarrow next \rightarrow next == head$
C. $head \rightarrow next == rear$ D. $head \rightarrow next \rightarrow next == rear$
5. 若允许表达式内多种括号混合嵌套,则为检查表达式中括号是否正确配对的算法,通常选用的辅助结构是 (A)
A. 栈 B. 线性表
C. 队列 D. 二叉排序树
6. 已知主串 $s = ADBADABBAAB$,模式串 $t = ADAB$,则应用朴素的串匹配算法进行模式匹配过程中,无效位移的次数是 (B)
A. 2 B. 3
C. 4 D. 5
7. 串 $s = Datastructure$ 中长度为 3 的子串的数目是 (C)
A. 9 B. 11
C. 12 D. 14
8. 假设以行优先顺序存储三维数组 $R[6][9][6]$,其中元素 $R[0][0][0]$ 的地址为 2100,且每个元素占 4 个存储单元,则存储地址为 2836 的元素是 (B)
A. $R[3][3][3]$ B. $R[3][3][4]$
C. $R[4][3][5]$ D. $R[4][3][4]$
9. 除第一层外,满二叉树中每一层结点个数是上一层结点个数的 (C)
A. 1/2 倍 B. 1 倍
C. 2 倍 D. 3 倍
10. 对于含 n 个顶点和 e 条边的图,采用邻接矩阵表示的空间复杂度为 (D)
A. $O(n)$ B. $O(e)$
C. $O(n+e)$ D. $O(n^2)$
11. 如果求一个连通图中以某个顶点为根的高度最小的生成树,应采用 (B)
A. 深度优先搜索算法 B. 广度优先搜索算法
C. 求最小生成树的 prim 算法 D. 拓扑排序算法
12. 快速排序在最坏情况下的时间复杂度是 (B)
A. $O(n^2 \log_2 n)$ B. $O(n^2)$
C. $O(n \log_2 n)$ D. $O(\log_2 n)$
13. 能进行二分查找的线性表,必须以 (A)
A. 顺序方式存储,且元素按关键字有序
B. 链式方式存储,且元素按关键字有序
C. 顺序方式存储,且元素按关键字分块有序
D. 链式方式存储,且元素按关键字分块有序
14. 为使平均查找长度达到最小,当由关键字集合 $\{05,11,21,25,37,40,41,62,84\}$ 构建二叉排序树时,第一个插入的关键字应为 (B)
A. 05 B. 37
C. 41 D. 62
15. ISAM 文件的周期性整理是为了空出 (D)
A. 磁道索引 B. 柱面索引

C . 柱面基本区 D . 柱面溢出区

二、填空题（本大题共 10 小题，每小题 2 分，共 20 分）

请在每小题的空格中填上正确答案。错填、不填均无分。

- 16 . 数据类型按其值能否分解 , 通常可分为 _____ 原子类型和结构类型 _____ 两种类型。
- 17 . 队列的修改是按 _____ 先进先出 _____ 的原则进行的。
- 18 . 两个串相等的充分必要条件是 两个串的长度相等且 _____ 相应位置上的字符相等 _____。
- 19 . 数组采用顺序存储方式表示是因为通常不对数组进行 _____ 插入和删除 _____ 操作。
- 20 . 用广义表的取表头 head 和取表尾 tail 的运算 , 从广义表 $LS=(b,c,(f),((d)))$ 中分解出原子 c 的操作为 _____head(tail(LS))_____。
- 21 . 结点数为 20 的二叉树可能达到的最大高度为 _____19_____。
- 22 . 带权连通图的生成树的权是该生成树上 _____ 各边的权值之和 _____。
- 23 . 所谓 “就地排序 ” , 是指排序算法辅助空间的复杂度为 _____ $O(1)$ _____ 的排序方法。
- 24 . 5 阶 B - 树的根结点至少含有 _____4_____ 个关键字。
- 25 . 索引文件中的索引表指示记录的关键字与 _____ 物理记录 _____ 之间一一对应的关系。

三、解答题（本大题共 4 小题，每小题 5 分，共 20 分）

26 . 假设以有序对 $\langle p,c \rangle$ 表示从双亲结点到孩子结点的一条边 , 若已知树中边的集合为 $\{ \langle a,b \rangle, \langle a,d \rangle, \langle a,c \rangle, \langle c,e \rangle, \langle c,f \rangle, \langle c,g \rangle, \langle c,h \rangle, \langle e,i \rangle, \langle e,j \rangle, \langle g,k \rangle \}$, 请回答下列问题 :

- (1) 哪个结点是根结点 ?
- (2) 哪些结点是叶子结点 ?
- (3) 哪些结点是 k 的祖先 ?
- (4) 哪些结点是 j 的兄弟 ?
- (5) 树的深度是多少 ?

27 . 已知有向图 G 的深度优先生成森林和广度优先生成森林如下。 请写出该图的深度优先遍历序列和广度优先遍历序列。

28 . 当将两个长度均为 n 的有序表 $A=(a_1, a_2, \dots, a_n)$ 与 $B=(b_1, b_2, \dots, b_n)$ ($a_i \leq b_j, 1 \leq i, j \leq n$) 归并为一个有序表 $C=(c_1, c_2, \dots, c_{2n})$ 时 , 所需进行的元素比较次数最少可达 n , 最多可达 $2n-1$ 。

(1) 假设有序表 $C=(2, 4, 5, 6, 7, 9)$, 试举出两组 A 与 B 的例子 , 使它们在归并过程中进行的元素比较次数分别达到最少和最多 ;

(2) 写出一般情况下 , 使归并所需进行的元素比较次数分别达到最少和最多时 , A 与 B 中的元素应满足的条件。

- (1)
- (2)

29 . 对下列关键字序列 (33 , 25 , 48 , 59 , 36 , 72 , 46 , 07 , 65 , 20) 构造表长为 19 的散列表。假设散列函数为 $h(\text{key})=\text{key}\%13$, 用开放地址法解决冲突 , 探查序列为 $d=h(\text{key}), d+12, d-12, \dots, d+22, d-22, d+32, \dots, d-32, \dots$ 等等。

- (1) 画出该散列表 ;
 - (2) 计算该散列表的装填因子 ;
 - (3) 求出等概率情况下查找成功的平均查找长度 ASL。
- (1) (2) (3)

四、算法阅读题（本大题共 4 小题，每小题 5 分，共 20 分）

30 . 已知 head 为带头结点的单循环链表的头指针 , 链表中的数据元素依次为 ($a_1, a_2, a_3, a_4, \dots, a_n$) , head 为指向空的顺序表的指针。阅读以下程序段 , 并回答问题 :

- (1) 写出执行下列程序段后的顺序表 A 中的数据元素 ;

(2) 简要叙述该程序段的功能。

```
if(head->next!=head)
{
p=head->next;
A->length=0;
while(p->next!=head)
{
p=p->next;
A->data[A->length ++]=p->data;
if(p->next!=head)p=p->next;
}
}
```

(1) $a_1, a_2, a_3, a_4, \dots$ 其中 n 为偶数时 $a_k = a_n$, 当 n 为奇数时 $a_k = a_{n-1}$

(2) 将链表中的数据元素 ($a_1, a_2, a_3, a_4, \dots$) 中的偶数项复制到 A 为指向的顺序表

31. 已知链串的存储结构描述如下：

```
#define NodeSize 4
typedef struct Node {
char data [NodeSize];
struct Node * next;
} * LinkStr;
```

阅读下列算法，并回答问题：

(1) t_1 和 t_2 的串值分别为 Chinese 和 China 时，写出 $f_{31}(t_1, t_2)$ 的返回值；

(2) t_1 和 t_2 的串值分别为 Japan 和 Japanes 时，写出 $f_{31}(t_1, t_2)$ 的返回值；

(3) t_1 和 t_2 的串值都为 string 时，写出 $f_{31}(t_1, t_2)$ 的返回值；

(4) 简述函数 f_{31} 的功能。

```
int f31(LinkStr t1, LinkStr t2)
{// 串值以 0 为结束符
int i;
while (1){
for (i=0; i<NodeSize; i++){
if (t1->data[i] == 0 && t2->data[i] == 0 return 0;
if (t1->data[i] == 0 ))return -1;
if (t2->data[i] == 0 ))return 1;
if (t1->data[i]>t2->data[i])return 1;
if (t1->data[i]<t2->data[i])return -1;
}
t1=t1->next;
t2=t2->next;
}
}
```

(1) 1

(2) -1

(3) 0

(4) 函数 f_3 的功能是比较两个字符串的大小，

当 $t_1=t_2$ 时，返回 0；当 $t_1>t_2$ 时返回 1；当 $t_1<t_2$ 时返回 -1；

32. 设二叉树采用二叉链表存储结构，结点的数据域 data 为字符类型。阅读下列算法，并回答问题：

(1) 对于如图所示的二叉树，写出执行函数 f32 的输出结果；

(2) 简述函数 f32 的功能。

```
void f32(BinTree T)
{ Stack s; // 定义栈 s
  BinTree p,q;
  if (T==NULL) return;
  InitStack(&s);
  p=T;
  do {
    while (p){
      Push(&s,p);
      if (p->lchild)p=p->lchild;
      else p=p->rchild;
    }
    while (!Stack Empty(s)&&q=StackTop(s)&&q->rchild==p){
      p=Pop(&s);
      printf("%c",p->data);
    }
    if(!StackEmpty(s)){
      q=StackTop(s);
      p=q->rchild;
    }
  } while (! Stack Empty(S));
}
```

(1) DBFGECA

(2) 后序遍历二叉树

33. 已知有向图的邻接表表示的形式说明如下：

```
#define Max Num    50          // 图的最大顶点数
typedef struct node {
  int adjvex;                // 邻接点域
  struct node * next;        // 链指针域
}EdgeNode; // 边表结点结构描述
typedef struct {
  char vertex;               // 顶点域
  EdgeNode *firstedge;       // 边表头指针
}VertexNode; // 顶点表结点结构描述
typedef struct{
  VertexNode adjlist [MaxNum]; // 邻接表
  int n,e;                   // 图中当前的顶点数和边数
}ALGraph; // 邻接表结构描述
```

下列函数 f33 是从有向图 G 中删除所有以 vi 为弧头的有向边。请在空缺处填入合适的内容，使

其成为一个完整的算法。

```
void f33 (ALGraph * G, int i)
{ int j;
  EdgeNode * p, *q;
  for (j=0;j<G->n;j= ++){
    p=G->adjlist [j].firstedge;
    while( (1) {
      q=p; p=p->next;
    }
    if(p!=NULL) {
      if (p !=G->adjlist[j].firstedge)q->next=p->next;
    }
    else( (2) ;
    free(p);
    G->e=( (3) ;
  }
}
```


(1) p->adjvex!=i&&p!=Null.
(2) p=p->next
(3) G->e-1

五、算法设计题（本大题 10 分）

34 . 在带头结点的循环链表 L 中，结点的数据元素为整型，且按值递增有序存放。给定两个整数 a 和 b，且 $a < b$ ，编写算法删除链表 L 中元素值大于 a 且小于 b 的所有结点。

第 34 题:在带头结点的循环链表 L 中，结点的数据元素为整型，且按值递增有序存放。给定两个整数 a 和 b，且 $a < b$ ，编写算法删除链表 L 中元素值大于 a 且小于 b 的所有结点。

```
void delete(linklist &l,int a,int b)
{pre=l; //pre 是前驱结点
  p=l->next; //p 是当前结点
  while(p!=l&&p->data<=a) // 直到遇到结点的数据域大于 a
  {pre=p;
    p=p->next;
  }
  while(p!=l&&p->data<b) // 删除大于 a 且小于 b 的结点 ,同时变换指针的指向
  {pre->next=p->next;
    free(p);
    p=p->next;
  }
}
```

 此题方法很多，这里的思路是先将当前指针 P 指到大于 a 的结点，然后在删除小于 b 的结点，如果遇到大于 B 的结点时，算法结束。

也可以用一个 WHILE 来删除，只不过当结点的值大于 B，算法不会结束，而是当前结点指到头指针为止，算法低效。

全国 2006 年 10 月高等教育自学考试

数据结构试题

课程代码： 02331

一、单项选择题（本大题共 15 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，请将其代码填写在题后的括号内。错选、多选或未选均无分。

1. 数据结构是（ ）

- A. 一种数据类型
- B. 数据的存储结构
- C. 一组性质相同的数据元素的集合
- D. 相互之间存在一种或多种特定关系的数据元素的集合

2. 算法分析的目的是（ ）

- A. 辨别数据结构的合理性
- B. 评价算法的效率
- C. 研究算法中输入与输出的关系
- D. 鉴别算法的可读性

3. 在线性表的下列运算中，不改变数据元素之间结构关系的运算是（ ）

- A. 插入 B. 删除
- C. 排序 D. 定位

4. 若进栈序列为 1, 2, 3, 4, 5, 6, 且进栈和出栈可以穿插进行，则可能出现的出栈序列为（ ）

- A. 3, 2, 6, 1, 4, 5 B. 3, 4, 2, 1, 6, 5
- C. 1, 2, 5, 3, 4, 6 D. 5, 6, 4, 2, 3, 1

5. 设串 $s_1 = \text{Data Structures with Java}$, $s_2 = \text{it}$ ，则子串定位函数 $\text{index}(s_1, s_2)$ 的值为（ ）

- A. 15 B. 16
- C. 17 D. 18

6. 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，则数组元素 $A[6][7]$ 的存储地址为（ ）

- A. 1207 B. 1209
- C. 1211 D. 1213

7. 在按层次遍历二叉树的算法中，需要借助的辅助数据结构是（ ）

- A. 队列 B. 栈
- C. 线性表 D. 有序表

8. 在任意一棵二叉树的前序序列和后序序列中，各叶子之间的相对次序关系（ ）

- A. 不一定相同 B. 都相同
- C. 都不相同 D. 互为逆序

9. 若采用孩子兄弟链表作为树的存储结构，则树的后序遍历应采用二叉树的（ ）

- A. 层次遍历算法 B. 前序遍历算法
- C. 中序遍历算法 D. 后序遍历算法

10. 若用邻接矩阵表示一个有向图，则其中每一列包含的 1 的个数为（ ）

- A. 图中每个顶点的入度 B. 图中每个顶点的出度
- C. 图中弧的条数 D. 图中连通分量的数目

11. 图的邻接矩阵表示法适用于表示（ ）

A . 无向图 B . 有向图

C . 稠密图 D . 稀疏图

12 . 在对 n 个关键字进行直接选择排序的过程中 , 每一趟都要从无序区选出最小关键字元素 , 则在进行第 i 趟排序之前 , 无序区中关键字元素的个数为 ()

A . i B . $i+1$

C . $n-i$ D . $n-i+1$

13 . 下列排序算法中 , 其时间复杂度和记录的初始排列无关的是 ()

A . 插入排序 B . 堆排序

C . 快速排序 D . 冒泡排序

14 . 若有序表的关键字序列为 ($b, c, d, e, f, g, q, r, s, t$) , 则在二分查找关键字 b 的过程中 , 先后进行比较的关键字依次为 ()

A . f, c, b B . f, d, b

C . g, c, b D . g, d, b

15 . 若在文件中查询年龄在 60 岁以上的男性及年龄在 55 岁以上的女性的所有记录 , 则查询条件为 ()

A . (性别 = “男”) OR (年龄 > 60) OR (性别 = “女”) OR (年龄 > 55)

B . (性别 = “男”) OR (年龄 > 60) AND (性别 = “女”) OR (年龄 > 55)

C . (性别 = “男”) AND (年龄 > 60) OR (性别 = “女”) AND (年龄 > 55)

D . (性别 = “男”) AND (年龄 > 60) AND (性别 = “女”) AND (年龄 > 55)

二、填空题 (本大题共 10 小题 , 每小题 2 分 , 共 20 分)

请在每小题的空格中填上正确答案。错填、不填均无分。

16 . 称算法的时间复杂度为 $O(f(n))$, 其含义是指算法的执行时间和 _____ 的数量级相同。

17 . 在一个长度为 n 的单链表 L 中 , 删除链表中 $*p$ 的前驱结点的时间复杂度为 _____。

18 . 假设为循环队列分配的向量空间为 $Q[20]$, 若队列的长度和队头指针值分别为 13 和 17 , 则当前尾指针的值为 _____。

19 . 设 $s = \text{I AM A ATHLETE}$, $t = \text{GOOD}$, 则执行下列串操作序列之后得到的 sub1 为 _____。

$\text{substr}(\text{sub1}, s, 5, 2)$; $\text{substr}(\text{sub2}, s, 6, 8)$; $\text{strcpy}(t1, t)$;

$\text{strcat}(t1, \text{sub2})$; $\text{strcat}(\text{sub1}, t1)$;

20 . 广义表的深度是指 _____。

21 . 一棵含 999 个结点的完全二叉树的深度为 _____。

22 . 含 n 个顶点的无向连通图中至少含有 _____ 条边。

23 . 对表长为 9000 的索引顺序表进行分块查找 , 假设每一块的长度均为 15 , 且以顺序查找确定块 , 则在各记录的查找概率均相等的情况下 , 其查找成功的平均查找长度为 _____。

24 . 若对关键字序列 (43 , 02 , 80 , 48 , 26 , 57 , 15 , 73 , 21 , 24 , 66) 进行一趟增量为 3 的希尔排序 , 则得到的结果为 _____。

25 . ISAM 文件由主索引、 _____、 _____ 和主文件组成。

三、解答题 (本大题共 4 小题 , 每小题 5 分 , 共 20 分)

26 . 某广义表的表头和表尾均为 ($a, (b, c)$) , 画出该广义表的图形表示。

27 . 已知二叉树的先序序列和中序序列分别为 HDACBGF 和 ADCBHFE G

(1) 画出该二叉树 ;

(2) 画出与 (1) 求得的二叉树对应的森林。

(1)

(2)

28. 已知带权图的邻接表如下所示，其中边表结点的结构为：

依此邻接表从顶点 C 出发进行深度优先遍历。

(1) 画出由此得到的深度优先生成树；

(2) 写出遍历过程中得到的从顶点 C 到其它各顶点的带权路径及其长度。

29. 从空树起，依次插入关键字 37, 50, 42, 18, 48, 12, 56, 30, 23, 构造一棵二叉排序树。

(1) 画出该二叉排序树；

(2) 画出从 (1) 所得树中删除关键字为 37 的结点之后的二叉排序树。

四、算法阅读题 (本大题共 4 小题，每小题 5 分，共 20 分)

30. 已知用有序链表存储整数集合的元素。阅读算法 f30，并回答下列问题：

(1) 写出执行 f30(a, b) 的返回值，其中 a 和 b 分别为指向存储集合 {2, 4, 5, 7, 9, 12} 和 {2, 4, 5, 7, 9} 的链表的头指针；

(2) 简述算法 f30 的功能；

(3) 写出算法 f30 的时间复杂度。

```
int f30(LinkList ha, LinkList hb)
{
    //LinkList 是带有头结点的单链表
    //ha 和 hb 分别为指向存储两个有序整数集合的链表的头指针
    LinkList pa, pb;
    pa = ha->next;
    pb = hb->next;
    while(pa && pb && pa->data == pb->data)
    {
        pa = pa->next;
        pb = pb->next;
    }
    if(pa == NULL && pb == NULL) return 1;
    else return 0;
}
```

(1)

(2)

(3)

31. 已知稀疏矩阵采用带行表的三元组表表示，其形式说明如下：

```
#define MaxRow 100 // 稀疏矩阵的最大行数

typedef struct {
    int i, j, v; // 行号、列号、元素值
}TriTupleNode;

typedef struct{
    TriTupleNode data[MaxSize];
    int RowTab[MaxRow+1]; // 行表
    int m, n, t; // 矩阵的行数、列数和非零元个数
}RTriTupleTable;
```

下列算法 f31 的功能是，以行优先的顺序输入稀疏矩阵的非零元 (行号、列号、元素值)，建立稀疏矩阵的带行表的三元组表存储结构。请在空缺处填入合适内容，使其成为一个完整的算法。

(注：矩阵的行、列下标均从 1 起计)

```

void f31(RTriTupleTable *R)
{ int i,k;
  scanf(   %d %d %d ,&R ->m,&R->n,&R->t);
  R->RowTab[1]=0;
  k=1;  //k      指示当前输入的非零元的行号
  for(i=0;                                ; i++)
  {   scanf(   %d %d %d ,                                ,                                ,&R->data[i].v);
      while(k<R->data[i].i)
      {
          ;
          R->RowTab[k]=i;
      }
  }
}

```

32 . 已知二叉树的存储结构为二叉链表，其类型定义如下：

```

typedef struct NodeType {
    DataType data;
    struct NodeType *lchild,*rchild;
}BinTNode,*BinTree;

```

阅读算法 F32，并回答下列问题：

(1) 对于如图所示的二叉树，画出执行算法 f32 的结果；

(2) 简述算法 f32 的功能。

```

BinTree f32(BinTree bt1)
{
    BinTree bt2;
    if(bt1==NULL)
        bt2=NULL;
    else {
        bt2=(BinTNode *)malloc(sizeof(BinTNode));
        bt2->data=bt1->data;
        bt2->rchild=f32(bt1->lchild);
        bt2->lchild=f32(bt1->rchild);
    }
    return bt2;
}

```

(1)

(2)

33 . 假设有向图采用邻接表表示法，其定义如下：

```

typedef struct {
    VertexNode adjlist[MaxVertexNum];
}

```



```
int n,e;          //          图的当前顶点数和弧数
} ALGraph;       //          邻接表类型
```

其中顶点表结点 VertexNode 结构为：

边表结点 EdgeNode 结构为：

下列算法 f33 的功能是，对以邻接表表示的有向图进行拓扑排序。

(1) 阅读算法 f33 ，并在空缺处填入合适的内容，使其成为一个完整的算法；

(2) 对于如图所示的邻接表，将执行算法 f33 后的 topo[] 结果填入给定的数组中。

```
void f33(ALGraph G, int topo [ ]){
    int i,j,k,count=0;
    int indegree[MaxVertexNum];
    EdgeNode *p;          //p          为指向边表结点的指针
    Queue Q;              //Q          为队列
    FindIndegree(G, indegree); //          求各顶点的入度，并置于入度向量 indegree
    InitQueue(&Q);
    for(i=0;i<G.n;i++)
        if(!indegree[i])EnQueue(&Q,i);
    while(!QueueEmpty(&Q)){
        j=                ;
        topo[j]=++count;
        for(p=G.adjlist[j].firstedge;p;p=p->next){
            k=p->adjvex;
            if(!(--indegree[k]))                ;
        }
    }
    if(count<G.          n)printf(          \n 图 G 中存在有环路  ) ;
}
```

(1)

(2) topo

五、算法设计题（本大题 10 分）

34 . 假设以带头结点的单链表表示有序表，单链表的类型定义如下：

```
typedef struct node{
    DataType data;
    struct node *next
}LinkNode, *LinkList;
```

编写算法，从有序表 A 中删除所有和有序表 B 中元素相同的结点。

全国2006年10月高等教育自学考试 数据结构试卷参考答案

课程代码: 02331

由华夏大地教育网老师提供

一、单项选择题 (本大题共15小题, 每小题2分, 共30分)

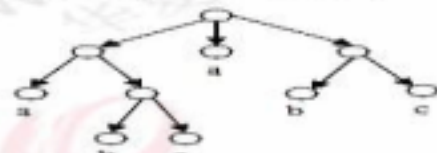
1. D 2. B 3. D 4. B 5. D
6. A 7. A 8. D 9. C 10. A
11. C 12. D 13. B 14. A 15. C

二、填空题 (本大题共10小题, 每小题2分, 共20分)

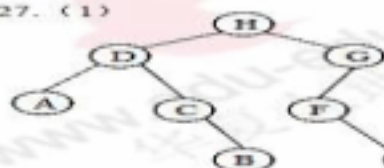
16. $f(n)$
17. $O(n)$
18. 10
19. $substr(sub1, s, 5, 2); substr(sub2, s, 6, 8); strcpy(t1, t); strcat(t1, sub2); strcat(sub1, t1);$
20. 表展开后所含括号的层数
21. 10
22. $n=1$
23. 509.5
24. (15, 02, 21, 24, 26, 57, 43, 66, 81, 48, 73)
25. 柱面索引、磁道索引

三、计算题 (本大题共5小题, 每小题5分, 共25分)

26. 该广义表为 $((a, (b, c)), a, (b, c))$



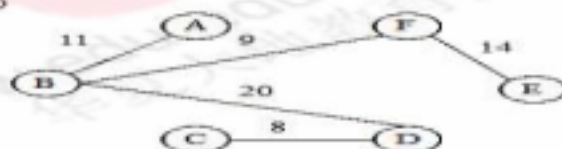
27. (1)



(2)

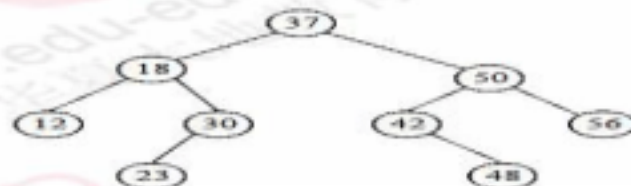


28. (1)

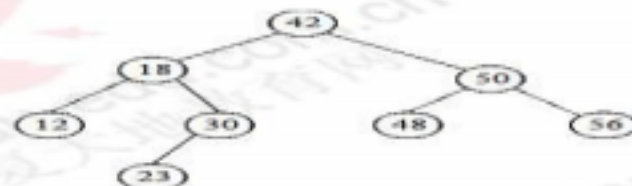


- (2) 顶点 C 到顶点 A 的带权路径为 (C, D, B, A), 其长度为 $8+20+11=39$
- 顶点 C 到顶点 B 的带权路径为 (C, D, B), 其长度为 $8+20=28$
- 顶点 C 到顶点 D 的带权路径为 (C, D), 其长度为 8
- 顶点 C 到顶点 E 的带权路径为 (C, D, B, F, E), 其长度为 $8+20+9+14=51$
- 顶点 C 到顶点 F 的带权路径为 (C, D, B, F), 其长度为 $8+20+9=37$

29. (1)



(2)



四、算法阅读题 (本大题共4小题, 每小题5分, 共20分)

30.

- (1) 0
- (2) 判断两个有序整数集合的链表是否相同 (结点个数相同, 对应结点的值相等), 若相同, 返回 1; 不同, 返回 0.
- (3) $O(n)$

31.

- (1) $i \leftarrow R \rightarrow t$
- (2) $\&R \rightarrow data[i].i$
- (3) $\&R \rightarrow data[i].j$
- (4) $k \leftarrow$

32.

(1)



- (2) 将二叉树 bt1 复制到 bt2。复制时, 将 bt1 的所有结点的左子树复制到 bt2 对应的右子树, 将 bt1 的所有结点的右子树复制到 bt2 对应的左子树。

33.

- (1) ① DeQueue(&Q) ② EnQueue(&Q, k)

- (2) topo

1	2	3	5	8	7	6	4
0	1	2	3	4	5	6	7

五、算法设计题 (本大题10分)

34.

```

Void deletelistlinklist (A, linklist B)
{
    linklist p, q, r;
    r=A;    p=A->next;    q=B->next;
    while(p!=NULL&&q!=NULL)
        if (p->data>q->data)
            q=q->next;
        else if (p->data<q->data)
            {r=p; p=p->next;}
        else
            {r->next=p->next;
             free(p);
             p=r->next;
            }
}
    
```