

数据结构试卷

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. 下列选项中与数据存储结构无关的术语是（ ）
A. 顺序表 B. 链表 C. 链队列 D. 栈
2. 将两个各有 n 个元素的有序表归并成一个有序表，最少的比较次数是（ ）
A. $n-1$ B. n C. $2n-1$ D. $2n$
3. 已知循环队列的存储空间大小为 m ，队头指针 $front$ 指向队头元素，队尾指针 $rear$ 指向队尾元素的下一个位置，则向队列中插入新元素时，修改指针的操作是（ ）
A. $rear=(rear-1)\%m$; B. $front=(front+1)\%m$;
C. $front=(front-1)\%m$; D. $rear=(rear+1)\%m$;
4. 递归实现或函数调用时，处理参数及返回地址，应采用的数据结构是（ ）
A. 堆栈 B. 多维数组 C. 队列 D. 线性表
5. 设有两个串 p 和 q ，其中 q 是 p 的子串，则求 q 在 p 中首次出现位置的算法称为（ ）
A. 求子串 B. 串联接 C. 串匹配 D. 求串长
6. 对于广义表 A ，若 $head(A)$ 等于 $tail(A)$ ，则表 A 为（ ）
A. $()$ B. $(())$ C. $((), ())$ D. $((), (), ())$
7. 若一棵具有 $n(n>0)$ 个结点的二叉树的先序序列与后序序列正好相反，则该二叉树一定是（ ）
A. 结点均无左孩子的二叉树 B. 结点均无右孩子的二叉树
C. 高度为 n 的二叉树 D. 存在度为 2 的结点的二叉树
8. 若一棵二叉树中度为 1 的结点个数是 3，度为 2 的结点个数是 4，则该二叉树叶子结点的个数是（ ）
A. 4 B. 5 C. 7 D. 8
9. 某算法有 3 个程序段，第一程序段的执行次数为 $2n^5$ ，第二个程序段执行次数为 $4n$ ，第三个程序段的执行次数为 $0.06n^2$ ，则该算法的时间复杂度为（ ）。
A. $O(n)$ B. $O(n \log_2 n)$ C. $O(n^5)$ D. $O(n^2)$
10. 已知有向图 $G=(V, E)$ ，其中 $V=\{V1, V2, V3, V4\}$ ， $E=\{\langle V1, V2 \rangle, \langle V1, V3 \rangle, \langle V2, V3 \rangle, \langle V2, V4 \rangle, \langle V3, V4 \rangle\}$ ，图 G 的拓扑序列是（ ）
A. $V1, V2, V3, V4$ B. $V1, V3, V2, V4$ C. $V1, V3, V4, V2$ D. $V1, V2, V4, V3$
11. 平均时间复杂度为 $O(n \log n)$ 的稳定排序算法是（ ）
A. 快速排序 B. 堆排序 C. 归并排序 D. 冒泡排序
12. 已知关键字序列为 (51, 22, 83, 46, 75, 18, 68, 30)，对其进行快速排序，第一趟划分完成后的关键字序列是（ ）
A. (18, 22, 30, 46, 51, 68, 75, 83) B. (30, 18, 22, 46, 51, 75, 83, 68)
C. (46, 30, 22, 18, 51, 75, 68, 83) D. (30, 22, 18, 46, 51, 75, 68, 83)
13. 某索引顺序表共有元素 395 个，平均分成 5 块。若先对索引表采用顺序查找，再对块中

元素进行顺序查找，则在等概率情况下，分块查找成功的平均查找长度是（ ）

- A.43 B.79 C.198 D.200

14.在含有 10 个关键字的 3 阶 B-树中进行查找，至多访问的结点个数为（ ）

- A.2 B.3 C.4 D.5

15. 设矩阵 A 是一个对称矩阵，为了节省空间，将其下三角部分按行优先存放在一维数组 B 中。对下三角矩阵中任一元素 a_{ij} (设矩阵 A 是一个对称矩阵，为了节省空间，将其下三角部分按行优先存放在一维数组 B 中，对下三角矩阵中任一元素 $a_{ij}(i \geq j)$ ，在一维数组 B 中下标 K 的值是()。

- A、 $i(i-1)/2+j-1$ B、 $i(i-1)/2+j$ C、 $i(i+1)/2+j-1$ D、 $i(i+1)/2+j$

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.数据结构由数据的逻辑结构、存储结构和数据的_____三部分组成。

17.在单链表中某结点后插入一个新结点，需要修改_____个结点指针域的值。

18.设栈 S 的初始状态为空，若元素 a、b、c、d、e、f 依次进栈，得到的出栈序列是 b、d、c、f、e、a，则栈 S 的容量至少是_____。

19.长度为零的串称为_____。

20.广义表 $G=(a,b,(c,d,(e,f)),G)$ 的长度为_____。

21.一棵树 T 采用孩子兄弟链表存储，如果树 T 中某个结点为叶子结点，则该结点在二叉链表中对应的结点一定是_____。

22.一个有 n 个顶点的无向连通图，最少有_____条边。

23.当待排关键字序列基本有序时，快速排序、简单选择排序和直接插入排序三种排序方法中，运行效率最高的是_____。

24.设目标串 $T="abccdcddccbaa"$ ，模式 $P="cdcc"$ 则第_____次匹配成功。若字符串的长度为 n,则子串的个数为_____

25.在归并排序中，若待排序记录的个数为 20，则共需要进行_____趟归并

三、解答题（本大题共 4 小题，每小题 5 分，共 20 分）

26.已知一棵二叉排序树（结点值大小按字母顺序）的前序遍历序列为 EBACDFHG，

请回答下列问题：

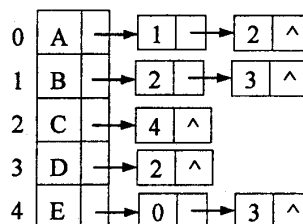
(1)画出此二叉排序树；

(2)若将此二叉排序树看作森林的二叉链表存储，请画出对应的森林。

27.已知有向图的邻接表如图所示，请回答下面问题：

(1)给出该图的邻接矩阵；

(2)从结点 A 出发，写出该图的深度优先遍历序列。

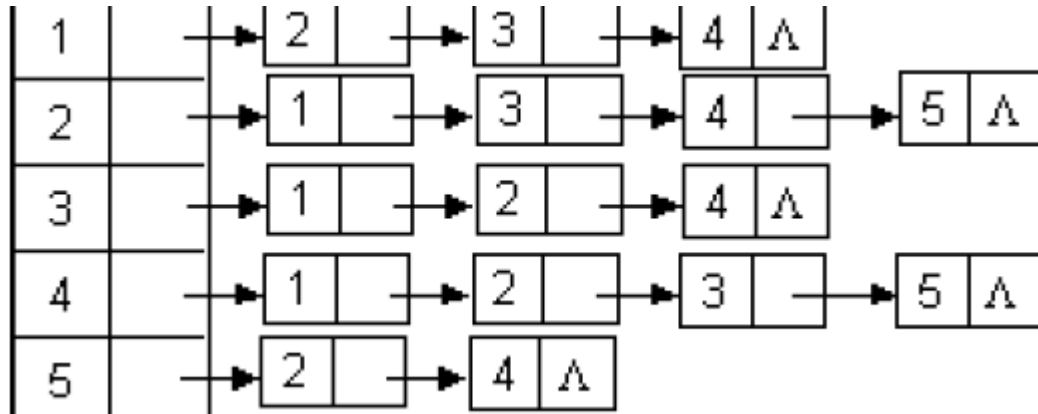


28. 在一个算法中需要建立多个堆栈是可以选用下列两种种方案之一，试问：这三种方

案之间相比较各有什么优缺点？

- (1) 分别用多个顺序存储空间建立多个独立的堆栈；
- (2) 多个堆栈共享一个顺序存储空间；

29. 设 $G=(V,E)$ 以邻接表存储，如图所示，试画出图的深度优先和广度优先生成图。



四、算法阅读题（本大题共 4 小题，每小题 5 分，共 20 分）

30. 阅读下列程序。

```

void f30(int A[], int n)
{
    int i,j,m; for (i=1; i<n; i++)
        for (j=0; j<i; j++)
        {
            m=A[i*n+j];
            A[i*n+j]=A[j*n+i];
            A[j*n+i]=m;
        }
}

```

回答下列问题：

(1) 已知矩阵 $B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ ，将其按行优先存于一维数组 A 中，给出执行函数调

用 $f30(A, 3)$ 后矩阵 B 的值；

31. 假设以二叉链表表示二叉树，其类型定义如下：

```

typedef struct node {
    char data;
    struct node* lchild, *rchild; // 左右孩子指针
}

```

```
    }    *BinTree;
```

阅读下列程序。

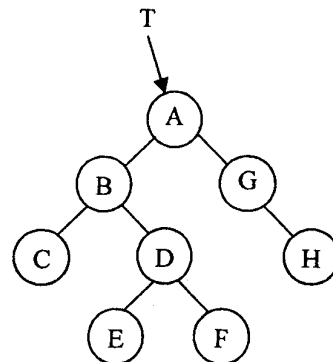
```
void f31(BinTree T)
{
    InitStack(S); // 初始化一个堆栈 S
    while (T != !StackEmpty(S))
    {
        while (T)
        {
            Push(S,T);    T=T->lchild;
        }
        if (!StackEmpty(S))
        {
            T=Pop(S); printf( "%c" ,T->data); T=T->rchild;
        }
    }
}
```

回答下列问题：

(1)已知以 T 为根指针的二叉树如图所示，

请写出执行 f31(T)的输出结果：

(2)简述算法 f31 的功能。



32.阅读下列程序。

```
void f32(int A[],int n)
{
    int i,j,m=l,t;
    for (i=0; i<n-l&&m; i++)
    {
        for (j=0; j<n; j++)
            printf( "%d " ,A[j]);
        printf( " \n" );
        m=0;
        for (j=1; j<n-i; j++)
            if (A[j-1]>A[j])
            {
```

```

        t=A[j-1];
        A[j-1]=A[j];
        A[j]=t;
        m=1;
    }
}
}

```

回答问题：

已知整型数组 A[]={34,26,15,89,42}，写出执行函数调用 f32(A,5)后的输出结果。

33. 程序填空

(1)

```

/////直接插入排序
void InsertSort(int a[],int n)
{
    int i,j,k=1;
    for(i=2;i<=n;i++)
    { if(a[i]<a[i-1])
        {
            a[0]=a[i];
            _____;
            for(j=i-2;a[0]<a[j];j--)
                a[j+1]=a[j];
            a[j+1]=a[0];}
        printf("\n 第%d 趟结果为:",i-1);
        for(k=1;____;k++)
            printf(" %d",a[k]);
    }
}

```

五、算法设计题（本题 10 分）

34.假设用带头结点的单循环链表表示线性表，单链表的类型定义如下：

```

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

```

编写程序，求头指针为 head 的单循环链表中 data 域值为正整数的结点个数占结点总数的比例，若为空表输出 0，并给出所写算法的时间复杂度。函数原型为：

```
float f34(LinkList head);
```

一、单项选择题

1—5 D B D A A

6—10 B C B C A

11—15 C D A B D

二、填空题

16、 运算

17、 2

18、 3

19、 空串

20、 4

21、 左右指针域均为空

22、 $n1$

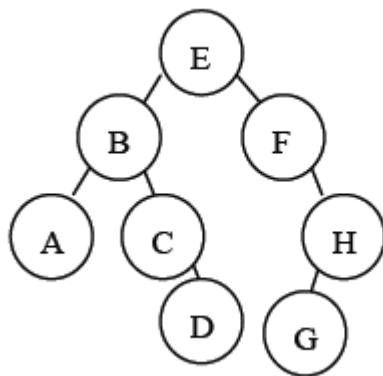
23、 直接插入排序

24、 $6 \quad n(n+1)/2$

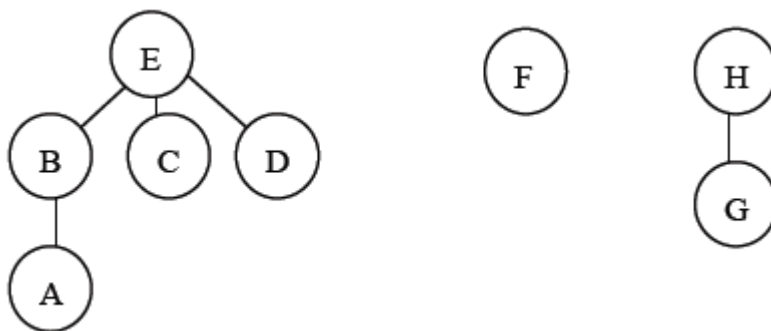
25、 5

三、解答题

26、 (1)



(2)



27、 该图的邻接矩阵如下：

0 1 1 0 0

0 0 1 1 0

0 0 0 0 1

0 0 1 0 0

1 0 0 1 0

(2)图的深度优先遍历序列如下：

A B C E D

28、(1) 每个栈仅用一个顺序存储空间时，操作简便，但分配存储空间小了，容易产生溢出，分配空间大了，容易造成浪费，各栈不能共享空间。

(2) 多个栈共享一个顺序存储空间，充分利用了存储空间，只有在整个存储空间都用完时才能产生溢出，其缺点是当一个栈满时要向左、右栈查询有无空闲单元。如果有，则要移动元素和修改相关的栈底和栈顶指针。当接近栈满时，查询空闲单元、移动元素和修改栈底栈顶指针的操作频繁，计算复杂并且耗费时间。

29、设从顶点 1 开始遍历，则深度优先生成树为

1---2---3---4---5

宽度优先生成树为

```
      1
     / \
    2  3  4
     \
      5
```

输出结果：V1 V4 V3 V6 V2 V5

四、算法阅读题

30、矩阵 B 的值为：

```
1 4 7
2 5 8
3 6 9
```

31、(1) 执行 f31(T)的输出结果为：

C B E D F A G H

(2)算法 f31 实现的功能是：利用栈实现二叉树的中序遍历。

32、执行函数调用 f32(A,5)后的输出结果是：

```
34,26,15,89,42
26,15,34,42,89
15,26, 34,42,89
```

33、 a[i]=a[i-1] k<=

五、算法设计题

34、解：

```
float f34(LinkList head)
```

```
{
```

```
    LinkList p;
```

```
    Int count=0, scount=0; //count 代表 data 域值为正整数的结点个数，scount
```

代表总结点个数。

```
    If(head>
```

```
        next==head) return 0;
```

```
    Else
```

```
    {
```

```
        P=head;
```

```
        Do
```

```
        { p=p->
```

```
next;
Scount++;
If(p>
data>0);
}
While(p>
next!=head);
}
Printf(“正整数的结点个数占结点总数的比例是:%d”,count/scount*1.0);
}
此算法的时间复杂度为：  $O(n)$ 
```


数据结构期末试卷

出卷人：09 数煤（1）班 1~9 号

一、判断题 (5 分)

- 1、线性表是一种随机存取结构----- ()
- 2、循环链表的特点是最后一个结点的指针域为 NULL----- ()
3. 下列不等式是否正确：
 $O(n) < O(n \log 2n) < O(n^2) < O(n^3) < O(nk) < O(2n)$ ----- ()
4. 数据元素之间的关系在计算机中有两种不同的表示方法：顺序映像和非顺序映像，分别对应两种不同的存储结构：链式存储结构和顺序存储结构----- ()
5. 抽象数据类型不可通过固有数据类型来表现和实现----- ()

二、选择题 (15 分)

- 1、对顺序存储的线性表，设其长度为 n ，在任何位置上插入或删除操作都是等概率的。插入一个元素时平均要移动表中的（ ）个元素。

(A) $n/2$ (B) $(n+1)/2$ (C) $(n-1)/2$ (D) n

- 2、一个栈的入栈序列是 1, 2, 3, 4, 5, 则栈的不可能的输出序列是 ()

A 3, 5, 4, 2, 1

B 3, 2, 4, 5, 1

C 1, 2, 3, 4, 5

D 5, 4, 3, 1, 2

- 3、已知广义表 $LS=((a,b,c),(d,e,f))$,运用 head 和 tail 函数取出 LS 中原子 e 的运算是()。

A. $\text{head}(\text{tail}(\text{LS}))$

B. tail(head(LS))

C. $\text{head}(\text{tail}(\text{head}(\text{tail}(\text{LS}))))$

D. head(tail(tail(head(LS))))

- 4、对稀疏矩阵进行压缩存储目的是 ()。

A. 便于进行矩阵运算 B. 便于输入和输出 C. 节省存储空间 D. 降低运算的时间复杂度

- 5、假设以行序为主序存储二维数组 $A=array[1..100, 1..100]$ ，设每个数据元素占 2 个存储单元，基地址为 10，则 $LOC[5, 5]=$ ()。

A. 808

B. 818

C. 1010

D. 1020

6. 在一个具有 n 个顶点的有向图中, 若所有顶点的出度之和为 s , 则所有顶点的入度之和为 ()

A. S

B.s-1

 C_{s+1}

D.n

- 7.若要把 n 个顶点连接为一个连通图, 则至少需要 () 条边

A. n

B.n+1

C.n-1

D.2n

8. 已知一个有向图的边集为 $\{ \langle a, b \rangle, \langle a, c \rangle, \langle b, d \rangle, \langle b, e \rangle, \langle d, e \rangle \}$, 则由该图产生的一种可能的拓扑序列为 ()

A. a,b,c,d,e

B.a,b,d,e,b

C.a,c,b,e,d

D.a,c,d,b,e

- 9.若在线性表中采用二分查找法查找元素,该线性表应该()

A. 元素按值有序

B.采用顺序存储结构

C. 元素按值有序，且采用顺序存储结构

D. 元素按值有序，且采用链式存储结构

- 10.二分查找法要求查找表中各元素的键值必须是（ ）排序

A. 递增或递减

- B. 递增
- C. 递减
- D. 无序

11. 采用顺序搜索方法查找长度为 n 的顺序表时，搜索成功的平均搜索长度为 ()

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

12. 适用于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. 三叉排序表 D. 线性链表

13. 设有序表中有 1000 个元素，则用二分查找查找元素 X 最多需要比较 () 次

- A. 25 B. 10 C. 7 D. 1

14. 下列对于循环队列的说法，正确的是： ()

- A 循环队列就是队列的顺序存储方式
- B 判断循环队列 Q 满的条件是： $Q.rear=Q.front$ (即队头指针与队尾指针值相同)
- C 判断循环队列 Q 满的条件是： $Q.rear=Q.front=0$
- D 循环队列的存储不要求用一组地址连续的存储单元

15. 在一个链栈中，已知 s 为栈顶指针 (直接指向栈顶元素结点，无头结点)， t 为栈底指针，直接指向栈底元素，则插入 r 结点的操作为： ()

A $t \rightarrow next=r; t=r;$

B $r \rightarrow next=s; s=r;$

C $s \rightarrow next=r; s=r;$

D $r \rightarrow next=t;$

三、填空题 (10 分)

1. 对于双向链表，在两个结点之间插入一个新结点需要修改的指针供_____个，单链表为_____个。

2. 判断队列是“空”还是“满”，可采用 2 中处理方法：1 是另设一个_____来区别 是满还是空：少用一个_____。

3. 广义表 $(a, (a, b), d, e, ((i, j), k))$ 的长度是_____，深度是_____。

4. 在一个具有 n 个顶点的无向完全图中，包含有_____条边，在一个具有 n 个顶点的有向完全图中，包含有_____条边。

5. 快速排序在平均情况下的时间复杂度为_____，在最坏情况下的时间复杂度为_____。

四、读程序写算法 (3*5 分)

1. void MergeList_L(LinkList & La, LinkList &Lb, LinkList &Lc)

```
{
    pa = La->next;
    pb = Lb->next;
    pc = Lc = Lb;
    while (pa && pb)
    {
        if (pa->data <= pb->data)
        {
            pc->next = pa;
            pc = pc->next;
            pc->next = pb;
            pa = pa->next;
```

```

    }
    else
    {
        pc->next =pb; pc = pc->next; pb= pb->next;
    }
} //end while
pc->next = pa?pa:pb;
free(La);
} //end function
2、 void YHT()
{
    SeqQueue Q;
    InitQueue(&Q);
    EnterQueue(&Q,1);
    For(n=2;n<=N;n++)
    {
        EnterQueue(&Q,1);
        for(i=1;i<=n-2;i++)
        {
            DeleteQueue(&Q,&temp);
            printf("%d",temp);
            GetHead(Q,&x);
            temp=temp+x;
            EnterQueue(&Q,temp);
        }
        DeleteQueue(&Q,&x);
        printf("%d",x);
        EnterQueue(&Q,1);
    }
}

```

五、程序填空（3*5 分）

1、已知不带头结点的线性链表 list，链表中结点构造为（data、link），其中 data 为数据域，link 为指针域。请完成下面算法填空，要求将该链表按结点数据域的值得大小从小到大重新链接。要求链接过程中不使用除该链表意外的任何链结点空间。

```

Linklist LinkListSort(linklist list)
{
    Linklist p,r,q;
    P=list->next;    //p 是工作指针，指向待排序的当前元素
    List->next=0;
    While(p)
    {
        r=p->next;    //r 是 p 的后继
        q=list;

```

```

If((____)>(____)) //处理待排序结点 p 比第一个元素结点小的情况
{
    P->next = list;
List = p;
}
Else //查找元素值最小的结点
{
while((____)&&(____))
    q=q->next;
    _____; //将当前排序结点链入有序链表中
    q->next=p;
}
p=r;
}
return p;
}

```

2、下列算法实现求采用顺序结构存储的串 s 和串 t 的一个最长公共子串。

```

void maxcomstr(orderstring *s,*t;int index,length)
{
    int i,j,k,length1,com;
    index=0;length=0;i=1;
    while(i<=s.len)
    {
        j=1;
        while(j<=t.len)
        { if(s[i]==t[j])
            { k=1;length1=q;con=1;
                while(con) if__(1)____{ length1=length1+1;k=k+1;}else__(2)____;
                if(length1>length){index=i;length=length1;}
                ____ (3) ____;
            }
            else__ (4) ____;
        }
        ____ (5) ____;
    }
}

```

3、Void preorder(bitree *T)

```

{
    bitree *stack[m];
    int top;
    if (T!=NULL)
    {
        top=1;
        stack[top]=(1);
    }
}

```

```

while( (2) )
{
    p=stack[top];
    top--;
    printf( "%d",p->data);
    if(p->rchild!=NULL)
    {
        ( 3 );
        stack[top]=p->rchild;}
    if( (4) )
    { top++;
        ( 5 );
    }
}
}

```

六、算法实现 (2*10 分)

1、已知整数数组 A，从第一个单元（即 A[1]）开始存储数据，且一共存储了 n 个元素。要求编写折半查找元素 e 的过程。当数组中存在元素 e 时，返回其下标，否则返回 0。（10 分）

Int BinarySearch(int *A,int n,int e)

2、已知二叉搜索树中的结点类型用 BTreeNode 表示，被定义为：

```
struct BTreeNode {ElemType data; BTreeNode *left, *right};
```

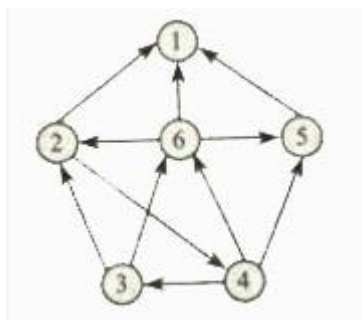
其中 data 为结点值域，left 和 right 分别为指向左、右孩子结点的指针域。假定具有 BTreeNode*类型的指针参数 BST 指向一棵二叉搜索树 的根结点，试根据下面的函数声明编写一个非递归算法，向 BST 树中插入值为 item 的结点，若树中不存在 item 结点则进行插入并返回 1 表示插入成功，若 树中已存在 item 结点则不插入并返回 0 表示插入失败。

七、简答题 (2*5 分)

1、如图，为一有向图，按要求回答问题：（5'）

写出各顶点的入度、出度和度。（2'）

写出该图的临接表。（3'）



2、有一个待排序的序列含 7 个记录，这 7 个关键字分别为 23,4,15,8,19, 24,15，用直接插入法对这个序列进行排序。

七、附加题：（10 分）

冒泡排序算法是把大的元素向上移（气泡的上浮），也可以把小的元素向下移（气泡的下沉），请给出上浮和下沉过程交替的冒泡排序算法。

参考答案:

一、1 对 2 错, 最后一个结点的指针域指向头结点 3 对 4 错 5 错

二、1、C 2、D 3、C 4、C 5、B 6、A 7、C 8、A 9、C 10、A
11、D 12、C 13、B 14、C 15、D

三、1、4 2 2、标志位, 元素空间 3、5 3 4、 $n(n-1)/2, n(n-1)$ 5、 $O(n\log 2n)$ $O(n^2)$

四、1、La Lb 带有头结点, 按非递减顺序排列, 归并 La Lb 得到 Lc, 要求 Lc 非递减顺序

2、算法功能: 打印杨辉三角的前 n 行元素

五、1、q->data

p->data

Q->next

Q->next->data<p->data

P->next=q->next

2、(1) $i+k \leq s.len \ \&\& \ j+k \leq t.len \ \&\& \ s[i+k] == t[j+k]$

(2) con=0 (3) j+=k (4) j++ (5) i++

3、(1) T

(2) top>0

(3) top++

(4) p->lchild!=NULL

(5) stack[top]=p->lchild

六、1、Int BinarySearch(int *A, int n, int e)

```
{
    int low, high, mid;
    low=1;
    high=n;          1 分
    while(1)
    {
        mid=(low+high)/2;  2 分
        if (A[mid]==e)      1 分
        {
            return mid;      1 分
        }
        else if (A[mid]>e)    1 分
        {
            high=mid-1;      1 分
        }
        else{
            low=mid+1; }      1 分
        if (low>high)        2 分
        return 0;}
}
```

2、int Insert (BTreeNode*& BST, const ElemType& item); //向二叉搜索树插入元素

```
{
    BTreeNode* t=BST, *parent=NULL;
    while(t!=NULL)
```

```

{
    parent=t;
    if(item==t->data) return 0;
    else if(item<t->data) t=t->left;
    else t=t->right;
} //建立值为 item, 左、右指针域为空的新结点
BTreeNode* p=new BTreeNode;
p->data=item;
p->left=p->right=NULL; //将新结点插入到二叉搜索树中的确定位置上
if(parent==NULL) BST=p;
else if(item<parent->data) parent->left=p;
else parent->right=p;
}
}

```

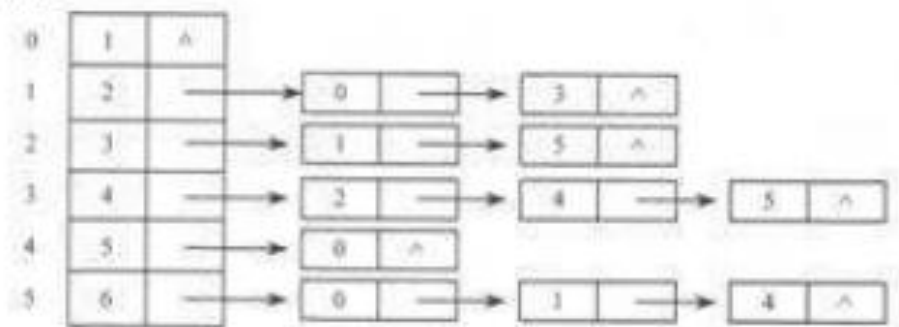
七、1、(1)

【解答】(1)各顶点入度、出度和度如下表所示。

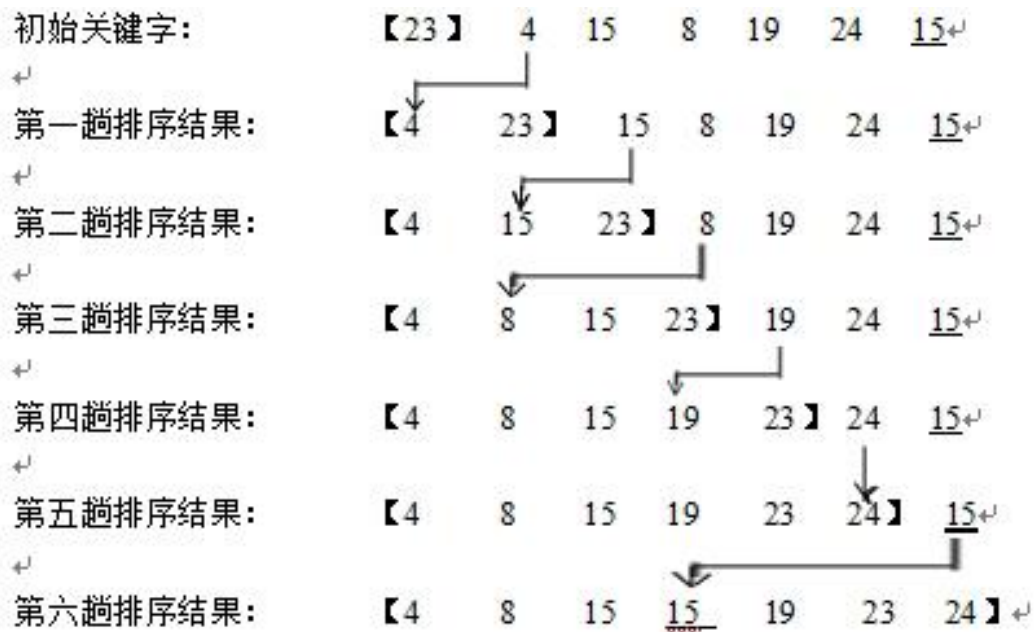
顶点	入度	出度	度
1	3	0	3
2	2	2	4
3	1	2	3
4	1	3	4
5	2	1	3
6	2	3	5

(2)

(3)邻接表如下所示。



2、【解】如图 方括号内的数据为已排好序的记录的关键字。在这个序列中有两个记录的关键字都为 15，为表示区别，将最后一个 15 加下划线。



附加题:

```
void BubbleSort2( int a[ ], int n)
```

```
{
    int i;
    int temp;
    int flag=1;
    int low=0;
    int high=n-1;
    while(low<high && flag)
    {
        flag=0;
        for(i=low;i<high;i++)
        {
            if(a[i]>a[i+1])
            {
                temp=a[i];
                a[i]=a[i+1];
                a[i+1]=temp;
                flag=1;
            }
        }
        high--;
        for(i=high;i>low;i--)
        {
            if(a[i]<a[i+1])
            {
                temp=a[i];
```



```
        a[i]=a[i+1];
        a[i+1]=temp;
        flag=1;
    }
}
low++;
}
```

一. 判断题

1. 堆排序是一种选择排序,堆实质上是一颗完全二叉树结点的层次序列。.....()
2. 串是多个字符组成的有限序列,其一般形式为 $S = \text{"a1a2.....an"}$ 。.....()
3. 二维数组的两种存储方式列序和行序。.....()
4. 数据结构是指相互之间存在一种或多种特定关系的数据元素的集合。.....()
5. 逻辑结构有线性结构,树形结构,图状结构三种。.....()

二. 填空题

1. 串的长度是指_____
2. 串是一种特殊的线性表,其特殊性体现在_____
3. 栈是_____的线性表,其运算遵循_____的原则。
4. 数组是一种_____数据结构,数组中的元素属于同一数据类型,一维数组可以看成向量,多维数组是向量的_____。
5. 二叉树是有 3 个基本单元组成____、____、_____
6. 深度为 k 的二叉树最多有_____个结点 ($k \geq 1$)

三. 选择题

1. 从逻辑上可以把数据结构分为()两大类。
A.动态结构、静态结构 B.顺序结构、链式结构
C.线性结构、非线性结构 D.初等结构、构造型结构
2. 连续储存设计时,储存单元的地址()
A.一定连续 B.一定不连续
C.不一定连续 D.部分连续,部分不连续
3. 一个栈的输出序列为 1,2,3,, n ,若输出序列的第一个元素是 n ,输出第 i ($1 \leq i \leq n$) 个元素是()
A. 不确定 B. $n-i+1$ C. i D. $n-i$
4. 执行完下列语句段后, i 值为:()。

```
int f(int x)
{return ((x>0) ? x* f(x-1):2); }
int i;
i = f(f(1));
```


A. 2 B. 4 C. 8 D. 无限递归
5. 下面关于串的叙述中,哪一个是不正确的?()
A.串是字符的有限序列
B.空串即是空白串
C.模式匹配是串的一种重要运算
D.串既可以采用顺序存储,也可以采用链式存储
6. 若串 $S = \text{"abdded"}$,其子串的数目是多少()
A.8 B.28 C.9 D.29
7. 假定有 K 个关键字互为同义词,若用线性探测法把这 K 个关键字存入散列表中,至少要进行多少次探测()
A. $K-1$ 次 B. K 次
C. $K+1$ 次 D. $K(K+1)/2$
1. 8. 下面四种排序方法中,不稳定的排序法是()。
A.插入 B.二路归并
C.冒泡 D.直接选择顺序

9. 下面程序段的时间复杂度为 ()

```
for (int i=0;i<m;i++)
    for (int j=0;j<n;j++)
        a[i][j]=i*j;
```

A $O(m^2)$ B $O(n^2)$ C $O(m*n)$ D $O(m+n)$

10. 若有一个 Queue 类型的队列 Q, 试问判断队列满的条件应是下列哪一个语句()

A `Q.front==Q.rear;` B `Q.front-Q.rear==Maxsize;`
C `Q.front+Q.rear==Maxsize;` D `Q.front==(Q.rear+1)%Maxsize;`

三. 算法题

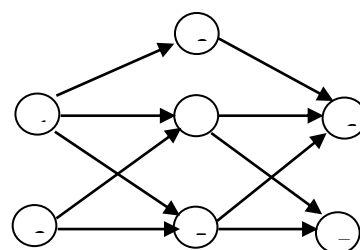
1. 设有序顺序表为 { 10, 20, 30, 40, 50, 60, 70 }, 采用折半搜索时, 搜索成功的平均搜索长度是多少?

2. 画出下列广义表的图形表示和它们的存储表示:

(1) D(A(c), B(e), C(a, L(b, c, d)))

(2) J1(J2(J1, a, J3(J1)), J3(J1))

3. 根据所给有向图, 写出一个拓扑序列。



五. 算法分析题

1、给出下列递归过程的执行结果

```
void unknown ( int w ) {
    if ( w ) {
        unknown ( w-1 );
        for ( int i = 1; i <= w; i++ ) cout << w<<' ' ;
        cout << endl;
    }
}
```

调用语句为 `unknown (4)`。

2. 设有一个二维数组 $A[m][n]$, 假设 $A[0][0]$ 存放在位置 $644_{(10)}$, $A[2][2]$ 存放在位置 $676_{(10)}$, 每个元素占一个空间, 问 $A[3][3]_{(10)}$ 存放在什么位置? 脚注 $_{(10)}$ 表示用 10 进制表示。

六. 简答题

在一个算法中需要建立多个堆栈是可以选用下列三种方案之一, 试问: 这三种方案之间相比较各有什么优缺点?

- (1) 分别用多个顺序存储空间建立多个独立的堆栈;
- (2) 多个堆栈共享一个顺序存储空间;
- (3) 分别建立多个独立的链接堆栈。

七. 附加题

写出二叉搜索树的删除算法。

一. 对 错 对 对 错

二. 串中所含字符的个数。

数据元素是字符

操作受限 先进后出

线性 推广

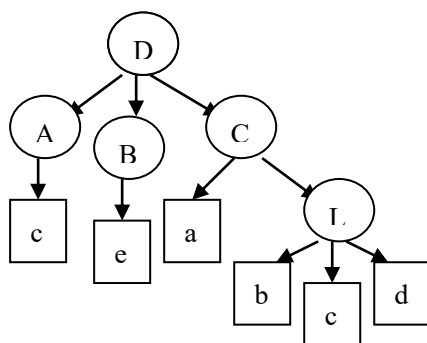
根结点 左子树 右子树

2 的 k 次方-1

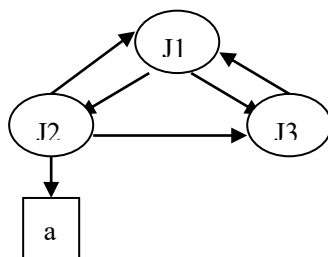
三. CABBB BDDCD

四 (1) 答案: $ASL_{succ} = (1*1 + 2*2 + 3*4) / 7 = 17 / 7$

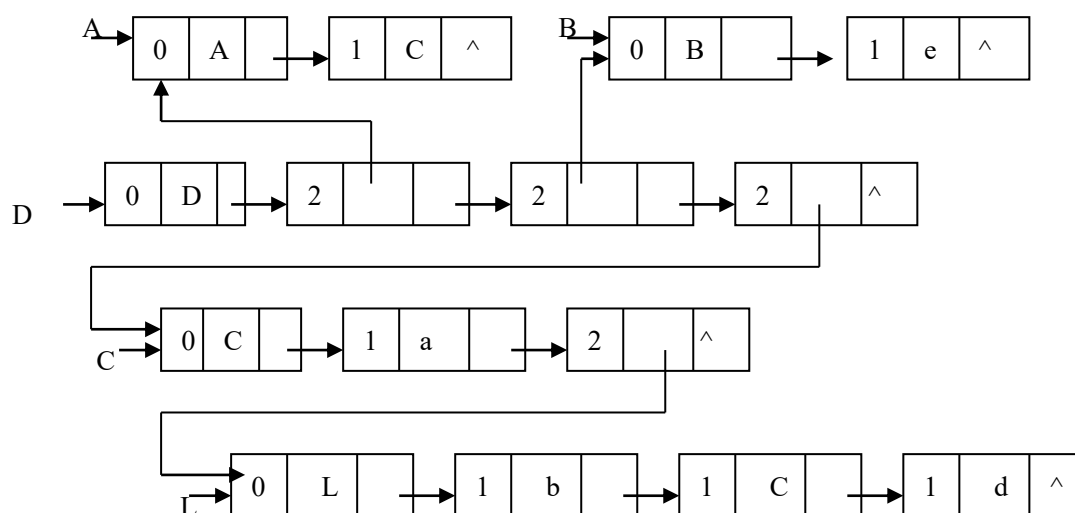
四 (2) 答案: 广义表 (1) 的图形表示为:



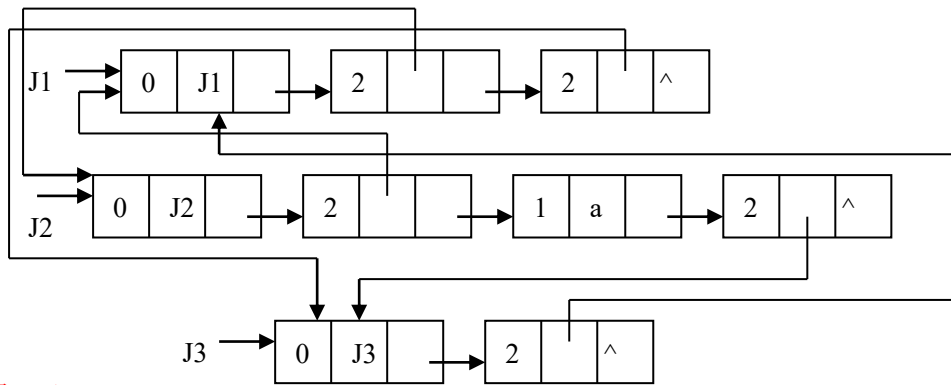
广义表 (2) 的图形表示为:



广义表 (1) 的存储表示为:



广义表 (2) 的存储表示为:



四 (3)

答案：其中的一个拓扑序列为：V1, V2, V3, V4, V5, V6, V7

五 (1)

答案：

(1) 1

2 2

3 3 3

4 4 4 4

五 (2)

答案：设数组元素 $A[i][j]$ 存放在起始地址为 $Loc(i, j)$ 的存储单元中。

因为： $Loc(2, 2) = Loc(0, 0) + 2*n + 2 = 644 + 2*n + 2 = 676$

所以： $n = (676 - 2 - 644) / 2 = 15$

所以： $Loc(3, 3) = Loc(0, 0) + 3*15 + 3 = 644 + 45 + 3 = 692$

六

答案：

(1) 每个栈仅用一个顺序存储空间时，操作简便，但分配存储空间小了，容易产生溢出，分配空间大了，容易造成浪费，各栈不能共享空间。

(2) 多个栈共享一个顺序存储空间，充分利用了存储空间，只有在整个存储空间都用完时才能产生溢出，其缺点是当一个栈满时要向左、右栈查询有无空闲单元。如果有，则要移动元素和修改相关的栈底和栈顶指针。当接近栈满时，查询空闲单元、移动元素和修改栈底栈顶指针的操作频繁，计算复杂并且耗费时间。

(3) 多个链栈一般不考虑栈溢出（仅受用户内存空间限制），缺点是栈中元素要以指针想链接，比顺序存储多占用了存储空间。

七

```
void bst_delete(bitreptr Tptr, KeyType key)
```

```
{
```

```
    BSTNode*parent=NULL, *p=Tptr, *q, *child;
```

```
    while(p)
```

```
    {
```

```

        if (p->key==key)

            break;

        parent=p;

        p=(key<p->key)? p->lchild: p->rchild;
    }

    if (!p)

        return;

    q=p;

    if (q->lchild&&q->rchild)

        for (parent=q. p=q->rchild; p->lchild; parent=p. p=p->lchild)

            child=(p->lchild)? p->lchild: p->rchild;

    if (!parent)

        Tptr=child;

    else

    {

        if (p==parent->lchild)

            parent->lchild=child

            if (p!=q)

                q->key=p->key;

    }

    free (p);
}

```