

第一章 概论

一、选择题

1、研究数据结构就是研究 (D)。

- A. 数据的逻辑结构
- B. 数据的存储结构
- C. 数据的逻辑结构和存储结构
- D. 数据的逻辑结构、存储结构

及其基本操作

2、算法分析的两个主要方面是 (A)。

- A. 空间复杂度和时间复杂度
- B. 正确性和简单性
- C. 可读性和文档性
- D. 数据复杂性和程序复杂性

3、具有线性结构的数据结构是 (D)。

- A. 图
- B. 树
- ~~C. 广义表~~
- D. 栈

4、计算机中的算法指的是解决某一个问题的有限运算序列，它必须具备输入、输出、(B)等5个特性。

- A. 可执行性、可移植性和可扩充性
- B. 可执行性、有穷性和确定性
- C. 确定性、有穷性和稳定性
- D. 易读性、稳定性和确定性

5、下面程序段的时间复杂度是 (C)。

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

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

$O(m+n)$

6、算法是 (D)。

- 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

7、某算法的语句执行频度为 $(3n+n\log_2 n+n^2+8)$ ，其时间复杂度表示 (C)。

- A. $O(n)$
- B. $O(n \log_2 n)$
- C. $O(n^2)$
- D. $O(\log_2 n)$

8、下面程序段的时间复杂度为 (C)。

```
i=1;
```

```
while(i<=n)
```

```
    i=i*3;
```

- A. $O(n)$
- B. $O(3n)$
- C. $O(\log_3 n)$
- D. $O(n^3)$

9、数据结构是一门研究非数值计算的程序设计问题中计算机的数据元素以及它们之间的 (B) 和运算等的学科。

- A. 结构
- B. 关系
- C. 运算
- D. 算法

10、下面程序段的时间复杂度是 (A)。

```
i=s=0;
```

```
while(s<n) {
```

```
    i++; s+=i;
```

```
}
```

- A. $O(\sqrt{n})$ B. $O(n^2)$ C. $O(\log_2 n)$
D. $O(n^3)$

11、抽象数据类型的三个组成部分分别为 (A)。

- A. 数据对象、数据关系和基本操作
B. 数据元素、逻辑结构和存储结构
C. 数据项、数据元素和数据类型
D. 数据元素、数据结构和数据类型

12、通常从正确性、易读性、健壮性、高效性等4个方面评价算法的质量，以下解释错误的是 (D)。

- A. 正确性算法应能正确地实现预定的功能
B. 易读性算法应易于阅读和理解，以便调试、修改和扩充
C. 健壮性当环境发生变化时，算法能适当地做出反应或进行处理，不会产生不需要的运行结果
D. 高效性即达到所需要的时间性能和空间

13、下列程序段的时间复杂度为 (B)。

```
x=n; y=0;
while (x >= (y+1) * (y+1))
    y=y+1;
```

$(x+1)^2 \leq n$
 $\sqrt{n} - 1$.

- A. $O(n)$ B. $O(\sqrt{n})$ C. $O(1)$ D. $O(n^2)$

二、填空题

1、程序段 “`i=1; while (i<=n) i=i*2;`” 的时间复杂度为 $\log_2 n$

n ? _____。

$\log_2 n$

2、数据结构的四种基本类型中，树形结构的元素是一对多关系。

线性表，栈和队，树，图

三、综合题

1、将数量级

$O(1)$, $O(N)$, $O(N^2)$, $O(N^3)$, $O(N \log_2 N)$, $O(\log_2 N)$, $O(2^N)$, $O(N!)$ 按增长率由小到大排序。

答案： $O(1)$ $O(\log_2 N)$ $O(N)$ $O(N \log_2 N)$ $O(N^2)$ $O(N^3)$ $O(2^N)$
 $O(N!)$

第二章 线性表

一、选择题

1、若长度为 n 的线性表采用顺序存储结构，在其第 i 个位置插入一个新元素算法的时间复杂度（ ）。

- A. $O(\log_2 n)$ B. $O(1)$ C. $O(n)$ D. $O(n^2)$

2、若一个线性表中最常用的操作是取第 i 个元素和找第 i 个元素的前趋元素，则采用（ ）存储方式最节省时间。*//随机访问：通过下标访问*

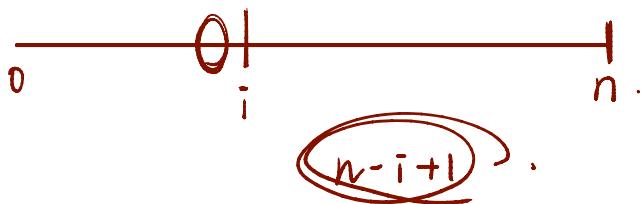
- A. 顺序表 B. 单链表 C. 双链表 D. 单循环链表

3、具有线性结构的数据结构是（ ）。

- A. 图 B. 树 C. 广义表 D. 栈

4、在一个长度为 n 的顺序表中，在第 i 个元素之前插入一个新元素时，需向后移动（ ）个元素。

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



5、非空的循环单链表 head 的尾结点 p 满足 ()。

- A. $p \rightarrow next == head$
- B. $p \rightarrow next == NULL$
- C. $p == NULL$
- D. $p == head$

6、链表不具有的特点是 ()。

- A. 可随机访问任一元素
- B. 插入删除不需要移动元素
- C. 不必事先估计存储空间
- D. 所需空间与线性表长度成正比

7、在双向循环链表中，在p指针所指的结点后插入一个指针q所指向的新结点，修改指针的操作是 ()。

A.

$p \rightarrow next = q; q \rightarrow prior = p; p \rightarrow next \rightarrow prior = q; q \rightarrow next = q;$

B.

$p \rightarrow next = q; p \rightarrow next \rightarrow prior = q; q \rightarrow prior = p; q \rightarrow next = p \rightarrow next;$

C.

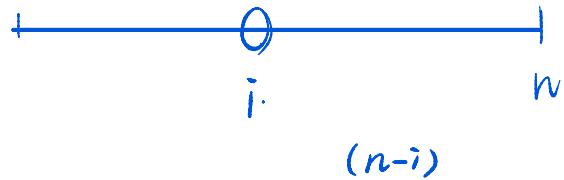
$q \rightarrow prior = p; q \rightarrow next = p \rightarrow next; p \rightarrow next \rightarrow prior = q; p \rightarrow next = q;$

D. $q \rightarrow next = p \rightarrow next; q \rightarrow prior = p; p \rightarrow next = q; p \rightarrow next = q;$

8、线性表采用链式存储时，结点的存储地址 ()。

- A. 必须是连续的
- B. 必须是不连续的
- C. 连续与否均可
- D. 和头结点的存储地址相连续

9、在一个长度为n的顺序表中删除第i个元素，需要向前移动 () 个元素。



A. $n-i$

B. $n-i+1$

C. $n-i-1$

D.

$i+1$

10、线性表是n个（ ）的有限序列。

A. 表元素

B. 字符

C. 数据元素

D. 数

据项

11、从表中任一结点出发，都能扫描整个表的是（ ）。

A. 单链表

B. 顺序表

C. 循环链表

D.

静态链表

12、在具有n个结点的单链表上查找值为x的元素时，其时间复杂度为（ ）。

A. $O(n)$

B. $O(1)$

C. $O(n^2)$

D. $O(n-1)$

13、线性表 $L=(a_1, a_2, \dots, a_n)$ ，下列说法正确的是（ ）。

A. 每个元素都有一个直接前驱和一个直接后继

B. 线性表中至少要有一个元素

C. 表中诸元素的排列顺序必须是由小到大或由大到小

D. 除第一个和最后一个元素外，其余每个元素都由一个且仅有一个

直接前驱和直接后继

90 92 94 96 98 100

14、一个顺序表的第一个元素的存储地址是90，每个元素的长度为2，则第6个元素的存储地址是（ ）。

A. 98

B. 100

C. 102

D. 106

15、在线性表的下列存储结构中，读取元素花费的时间最少的是（ ）。

A. 单链表

B. 双链表

C. 循环链表

D. 顺

$$P \rightarrow \text{next} = P \rightarrow \text{next} \rightarrow \text{next}$$

序表

16、在一个单链表中，若删除p所指向结点的后续结点，则执行（ ）。

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

17、将长度为n的单链表连接在长度为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. $O(m+n)$

18、线性表的顺序存储结构是一种（ ）存储结构。

- A. 随机存取
- B. 顺序存取
- C. 索引存取
- D. 散列存取

19、顺序表中，插入一个元素所需移动的元素平均数是（ ）。

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

10、循环链表的主要优点是（ ）。

- A. 不再需要头指针
- B. 已知某结点位置后能容易找到其直接前驱
- C. 在进行插入、删除运算时能保证链表不断开
- D. 在表中任一结点出发都能扫描整个链表

11、不带头结点的单链表 head 为空的判定条件是（ A ）。

- A. `head==NULL`
- B.
- `head->next==NULL`
- C. `head->next==head`
- D. `head!=NULL`

答案 B 是带头结点的

12、在下列对顺序表进行的操作中，算法时间复杂度为 $O(1)$ 的是（ ）。

A. 访问第*i*个元素的前驱 ($1 < i \leq n$) B. 在第*i*个元素之后插入一个新元素 ($1 \leq i \leq n$)

C. 删 除第*i*个元素 ($1 \leq i \leq n$) D. 对顺序表中元素进行排序

答案是 A.

假设顺序表 L，长度为 n，求第 *i* 个节点 L[i]，直接前驱 L[i-1]，因此为 0
(1)

答案 B 需要移动 $n-i+1$ 个节点，因此为 $O(n)$

答案 C 也需要移动 $n-i$ 个节点

答案 D 根据排序方法不同最慢 $O(n^2)$ ，最快 $O(n \log n)$

13、已知指针 p 和 q 分别指向某单链表中第一个结点和最后一个结点。假设指针 s 指向另一个单链表中某个结点，则在 s 所指结点之后插入上述链表应执行的语句为（ ）。

A. ~~q->next=s->next; s->next=p;~~

B. s->next=p; q->next=s->next;

C. p->next=s->next; s->next=q;

D. s->next=q; p->next=s->next;

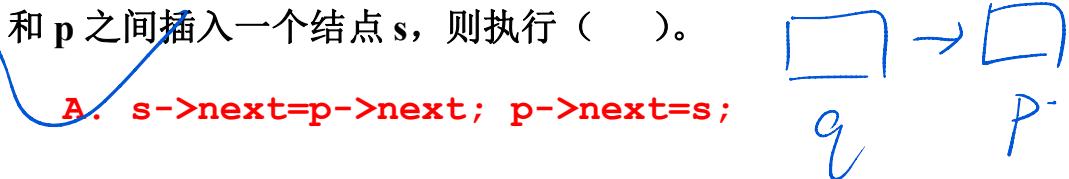
14、在以下的叙述中，正确的是（ ）。

- A. 线性表的顺序存储结构优于链表存储结构
- B. 线性表的顺序存储结构适用于频繁插入/删除数据元素的情况
- C. 线性表的链表存储结构适用于频繁插入/删除数据元素的情况
- D. 线性表的链表存储结构优于顺序存储结构

15、在表长为 n 的顺序表中，当在任何位置删除一个元素的概率相同时，
删除一个元素所需移动的平均个数为（ ）。

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

16、在一个单链表中，已知 q 所指结点是 p 所指结点的前驱结点，若在 q
和 p 之间插入一个结点 s，则执行（ ）。



- A. $s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s;$
- B. $p \rightarrow \text{next} = s \rightarrow \text{next}; s \rightarrow \text{next} = p;$
- C. $q \rightarrow \text{next} = s; s \rightarrow \text{next} = p;$
- D. $p \rightarrow \text{next} = s; s \rightarrow \text{next} = q;$

17、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
()。

- A. $p = p \rightarrow \text{next};$
- B. $p \rightarrow \text{next} = p \rightarrow \text{next} \rightarrow \text{next};$
- C. $p \rightarrow \text{next} = p;$
- D. $p = p \rightarrow \text{next} \rightarrow \text{next};$

18、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 $p \rightarrow next \rightarrow next == head$ ，则（ ）。

- 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

二、填空题

1、设单链表的结点结构为 (data, next)。已知指针p指向单链表中的结点，q指向新结点，欲将q插入到p结点之后，则需要执行的语句： $q \rightarrow next = p \rightarrow next$ $p \rightarrow next = q$ 。

答案： $q \rightarrow next = p \rightarrow next$ $p \rightarrow next = q$

2、线性表的逻辑结构是 线性结构，其所含元素的个数称为线性表的 长度。

答案： 线性结构 长度

3、写出带头结点的双向循环链表 L 为空表的条件 _____。

答案： $L \rightarrow prior == L \rightarrow next == L$

4、带头结点的单链表 head 为空的条件是 _____。

答案： $head \rightarrow next == NULL$

5、在一个单链表中删除p所指结点的后继结点时，应执行以下操作：

$q = p \rightarrow next;$

$p \rightarrow next = q \rightarrow next;$

三、判断题

- 1、单链表不是一种随机存储结构。 ✓
- 2、在具有头结点的单链表中，头指针指向链表的第一个数据结点。 ✗
- 3、用循环单链表表示的链队列中，可以不设队头指针，仅在队尾设置队尾指针。 ✓
- 4、顺序存储方式只能用于存储线性结构。 ✗
- 5、在线性表的顺序存储结构中，逻辑上相邻的两个元素但是在物理位置上不一定是相邻的。 ✗
- 6、链式存储的线性表可以随机存取。 ✗

四、程序分析填空题

- 1、函数GetElem实现返回单链表的第*i*个元素，请在空格处将算法补充完整。

```
int GetElem(LinkList L, int i, Elemtyp *e) {  
    LinkList p; int j;  
    p=L->next; j=1;  
    while (p&&j<i) {  
        p=p->next; ++j;  
    }  
    if (!p||j>i) return ERROR;
```

```

    *e = p->data ;
    return OK;
}

```

答案: (1) p=p->next (2) p->data

2、函数实现单链表的插入算法，请在空格处将算法补充完整。

```

int ListInsert(LinkList L, int i, ElemType e) {
    LNode *p, *s; int j;
    p=L; j=0;

    while ((p!=NULL) && (j<i-1)) {
        p=p->next; j++;
    }

    if (p==NULL || j>i-1) return ERROR;
    s=(LNode *)malloc(sizeof(LNode));
    s->data=e;

    s->next=p->next;
    p->next=s. ;
    return OK;
} /*ListInsert*/

```

答案: (1) s->next=p->next (2) p->next=s

3、函数ListDelete_sq实现顺序表删除算法，请在空格处将算法补充完整。

```

int ListDelete_sq(Sqlist *L, int i) {

```

```

int k;

if(i<1||i>L->length) return ERROR;

for(k=i-1;k<L->length-1;k++)

    L->slist[k]= _____ (1) _____ ;

    _____ (2) _____ ;

return OK;

}

```

答案: (1) L->slist[k+1] (2) --L->Length

4、函数实现单链表的删除算法，请在空格处将算法补充完整。

```

int ListDelete(LinkList L,int i,ElemType *s){

LNode *p,*q;

int j;

p=L;j=0;

while((_____ (1) _____)&&(j<i-1)) {

    p=p->next;j++;

}

if(p->next==NULL||j>i-1) return ERROR;

q=p->next;

_____ (2) _____;

*s=q->data;

free(q);

return OK;

```

```
 }/*listDelete*/
```

答案: (1) $p \rightarrow next \neq NULL$ (2) $p \rightarrow next = q \rightarrow next$

5、写出算法的功能。

```
int L(head) {  
    node * head;  
    int n=0;  
    node *p;  
    p=head;  
    while (p !=NULL)  
    { p=p->next;  
        n++;  
    }  
    return (n);  
}
```

答案: 求单链表head的长度

五、综合题

1、编写算法，实现带头结点单链表的逆置算法。

答案: void invert(Lnode *head)

```
{Lnode *p,*q;
```

```
if(!head->next) return ERROR;  
p=head->next; q=p->next; p->next=NULL;  
while(q)  
{p=q; q=q->next; p->next=head->next; head->next=p;}  
}
```

2、有两个循环链表，链头指针分别为 L1 和 L2，要求写出算法将 L2 链表链到 L1 链表之后，且连接后仍保持循环链表形式。

答案： void merge(Lnode *L1, Lnode *L2)

```
{Lnode *p,*q ;  
while(p->next!=L1)  
p=p->next;  
while(q->next!=L2)  
q=q->next;  
q->next=L1; p->next =L2;  
}
```

3、设一个带头结点的单向链表的头指针为 head，设计算法，将链表的记录，按照 data 域的值递增排序。

答案： void assending(Lnode *head)

```
{Lnode *p,*q , *r, *s;  
p=head->next; q=p->next; p->next=NULL;  
while(q)  
{r=q; q=q->next;
```

```
if(r->data<=p->data)

{r->next=p; head->next=r; p=r; }

else

{while(!p && r->data>p->data)

{s=p; p=p->next; }

r->next=p; s->next=r; }

p=head->next; }

}
```

4、编写算法,将一个头指针为 `head` 不带头结点的单链表改造为一个单向循环链表，并分析算法的时间复杂度。

答案：

```
void linklist_c(Lnode *head)

{Lnode *p; p=head;

if(!p) return ERROR;

while(p->next!=NULL)

p=p->next;

p->next=head;

}
```

设单链表的长度（数据结点数）为 N ，则该算法的时间主要花费在查找链表最后一个结点上（算法中的 `while` 循环），所以该算法的时间复杂度为 $O(N)$ 。

5、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 $(a_1, a_2, a_3, a_4, \dots, a_n)$, A 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;
    }
}
```

答案：

- (1) (a_2, a_4, \dots) (2) 将循环单链表中偶数结点位置的元素值写入顺序表 A
- 6、设顺序表 va 中的数据元数递增有序。试写一算法，将 x 插入到顺序表的适当位置上，以保持该表的有序性。

答案：

```
void Insert_sq(Sqlist va[], ElemtType x)
```

```

{int i, j, n;

n=length(va[]);

if(x>=va[i])

    va[n]=x;

else

    {i=0;

    while(x>va[i]) i++;

    for(j=n-1;j>=I;j--)

        va[j+1]=va[j];

    va[i]=x; }

n++;

}

```

7、假设线性表采用顺序存储结构，表中元素值为整型。阅读算法f2，设
顺序表L=(3, 7, 3, 2, 1, 1, 8, 7, 3)，写出执行算法f2后的线性表L的数
据元素，并描述该算法的功能。

```

void f2 (SeqList *L) {

    int i, j, k;

    k=0;

    for (i=0; i<L->length; i++) {

        for (j=0; j<k && L->data[i] !=L->data[j]; j++) {

            if (j==k) {

                if (k!=i) L->data[k]=L->data[i];

```

```
    k++ ;  
  
}  
  
}  
  
L->length=k;  
}
```

答案：

(3,7,2,1,8) 删除顺序表中重复的元素

8、已知线性表中的元素以值递增有序排列，并以单链表作存储结构。试写一算法，删除表中所有大于 x 且小于 y 的元素（若表中存在这样的元素）同时释放被删除结点空间。

答案：

```
void Delete_list(Lnode *head, ElemType x, ElemType y)  
{Lnode *p, *q;  
if(!head) return ERROR;  
p=head; q=p;  
while(!p)  
{if(p->data>x) && (p->data<y)}i++;  
if(p==head)  
{head=p->next; free(p);  
p=head; q=p; }
```

```
else
{q->next=p->next; free(p);
p=q->next; }

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

}
```

9、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。

第三章 栈和队列

一、选择题

1、一个栈的输入序列为： a, b, c, d, e，则栈的不可能输出的序列是（ ）。

A. a,b,c,d,e

B. d,e,c,b,a

C. ~~d,c,e,a,b~~

D. e,d,c,b,a

2、判断一个循环队列Q（最多n个元素）为满的条件是（ ）。

A. Q->rear==Q->front

B.

Q->rear==Q->front+1

C. ~~Q->front==(Q->rear+1)%n~~

D.

Q->front==(Q->rear-1)%n

3、设计一个判别表达式中括号是否配对的算法，采用 (D) 数据结构最佳。

A. 顺序表

B. 链表

C. 队列

D. 栈

4、带头结点的单链表head为空的判定条件是（ ）。

A. head==NULL

B. ~~head->next==NULL~~

C. head->next!=NULL

D. head!=NULL

5、一个栈的输入序列为：1,2,3,4，则栈的不可能输出的序列是（ ）。

A. 1243

B. 2134

C. 1432

D. ~~4312~~

E. 3214

6、若用一个大小为6的数组来实现循环队列，且当rear和front的值分别为0, 3。当从队列中删除一个元素，再加入两个元素后，rear和front的值分别为（ ）。

A. 1和5

B. ~~2和4~~

C. 4和2

D. 5

和1

7、队列的插入操作是在 (A)。

A. 队尾

B. 队头

C. 队列任意位置

D. 队

头元素后

8、循环队列的队头和队尾指针分别为`front`和`rear`, 则判断循环队列

为空的条件是 (A)。

A. `front==rear`

B. `front==0`

C. `rear==0`

D. `front=rear+1`

9、一个顺序栈`s`, 其栈顶指针为`top`, 则将元素`e`入栈的操作是 ()。

A. `*s->top=e; s->top++;`

B.

`s->top++; *s->top=e;`

C. `*s->top=e`

D. `s->top=e;`

10、表达式`a*(b+c)-d`的后缀表达式是 (B)。

A. `abcd+-`

B. `abc+*d-`

C. `abc*+d-`

D.

`-+*abcd`

11、将递归算法转换成对应的非递归算法时, 通常需要使用 () 来保存中间结果。

A. 队列

B. 栈

C. 链表

D. 树

12、栈的插入和删除操作在 (B)。

A. 栈底

B. 栈顶

C. 任意位置

D.

指定位置

13、五节车厢以编号1, 2, 3, 4, 5顺序进入铁路调度站(栈), 可以得到 () 的编组。

A. 3, 4, 5, 1, 2

B. 2, 4, 1, 3, 5

C. 3, 5, 4, 2, 1

D. 1, 3, 5, 2, 4

14、判定一个顺序栈 s (栈空间大小为 n) 为空的条件是 ()。

- A. $s \rightarrow top == 0$ B. $s \rightarrow top != 0$
C. $s \rightarrow top == n$ D. $s \rightarrow top != n$

15、在一个链队列中, $front$ 和 $rear$ 分别为头指针和尾指针, 则插入一个结点 s 的操作为 ()。

- A. $front = front \rightarrow next$ B. $s \rightarrow next = rear; rear = s$
C. $rear \rightarrow next = s; rear = s;$ D.
 $s \rightarrow next = front; front = s;$

16、一个队列的入队序列是 1, 2, 3, 4, 则队列的出队序列是 (A)。

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

17、依次在初始为空的队列中插入元素 a, b, c, d 以后, 紧接着做了两次删除操作, 此时的队头元素是 (C)。

- A. a B. b C. c D. d

18、正常情况下, 删除非空的顺序存储结构的堆栈的栈顶元素, 栈顶指针 top 的变化是 (D)。

- A. top 不变 B. $top = 0$ C. $top = top + 1$ D.

$top = top - 1$

19、判断一个循环队列 Q (空间大小为 M) 为空的条件是 ()。

- A. $Q \rightarrow front == Q \rightarrow rear$ B.
 $Q \rightarrow rear - Q \rightarrow front - 1 == M$

- C. $Q \rightarrow front + 1 == Q \rightarrow rear$ D.

$Q \rightarrow rear + 1 = Q \rightarrow front$

20、设计一个判别表达式中左右括号是否配对出现的算法，采用 (C) 数据结构最佳。

A. 线性表的顺序存储结构 B. 队列 C. 栈 D. 线性表的链式存储结构

21、当用大小为N的数组存储顺序循环队列时，该队列的最大长度为 ()。

A. N B. N+1 C. N-1 D. N-2

22、队列的删除操作是在 (A)。

A. 队首 B. 队尾 C. 队前 D. 队后

23、若让元素1, 2, 3依次进栈，则出栈次序不可能是 ()。

A. 3, 2, 1 B. 2, 1, 3 C. 3, 1, 2 D. 1, 3,

2

24、循环队列用数组A[0, m-1]存放其元素值，已知其头尾指针分别是 front和rear，则当前队列中的元素个数是 ()。

A. $(rear-front+m) \% m$ B. $rear-front+1$

C. $rear-front-1$ D. $rear-front$

25、在解决计算机主机和打印机之间速度不匹配问题时，通常设置一个打印数据缓冲区，主机将要输出的数据依次写入该缓冲区，而打印机则从该缓冲区中取走数据打印。该缓冲区应该是一个 (B) 结构。

A. 堆栈 B. 队列 C. 数组 D. 线性表

26、栈和队列都是 (C)。

A. 链式存储的线性结构

B. 链式存储的非线性结构

C. 限制存取点的线性结构

D. 限制存取点的非线性结构

27、在一个链队列中，假定front和rear分别为队头指针和队尾指针，删除一个结点的操作是（ ）。

A. front=front->next

B. rear= rear->next

C. rear->next=front

D. front->next=rear

28、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

二、填空题

1、设栈S和队列Q的初始状态为空，元素e1, e2, e3, e4, e5, e6依次通过栈S，一个元素出栈后即进入队列Q，若6个元素出队的序列是e2, e4, e3, e6, e5, e1，则栈的容量至少应该是 3。

答案：3

2、一个循环队列Q的存储空间大小为M，其队头和队尾指针分别为front和rear，则循环队列中元素的个数为：(尾-头+m)%m

答案：(rear-front+M)%M

3、在具有n个元素的循环队列中，队满时具有 n-1 个元素。

答案: n-1

4、设循环队列的容量为70, 现经过一系列的入队和出队操作后, front为20, rear为11, 则队列中元素的个数为61。

答案: 61

$$(20+1+70) \% 70.$$

5、已知循环队列的存储空间大小为20, 且当前队列的头指针和尾指针的值分别为8和3, 且该队列的当前的长度为_____。

三、判断题

1、栈和队列都是受限的线性结构。✓

2、在单链表中, 要访问某个结点, 只要知道该结点的地址即可; 因此, 单链表是一种随机存取结构。✗

3、以链表作为栈的存储结构, 出栈操作必须判别栈空的情况。✓

四、程序分析填空题

1、已知栈的基本操作函数:

```
int InitStack(SqStack *S); //构造空栈  
int StackEmpty(SqStack *S); //判断栈空  
int Push(SqStack *S, ELEMTYPE e); //入栈  
int Pop(SqStack *S, ELEMTYPE *e); //出栈
```

函数conversion实现十进制数转换为八进制数, 请将函数补充完整。

```
void conversion()  
{  
    InitStack(S);
```

```

scanf("%d", &N) ;

while (N) {
    Push(1, S, &N%8)
    N=N/8;

}

while ( | (2)(S) ) {
    Pop(S, &e) ;
    printf("%d", e) ;

}

}//conversion

```

答案: (1) Push(S, N%8) (2) !StackEmpty(S)

2、写出算法的功能。

```

int function(SqQueue *Q, ElemType *e) {

if (Q->front==Q->rear)

    return ERROR;

*e=Q->base[Q->front] ;

Q->front=(Q->front+1)%MAXSIZE;

return OK;

}

```

3、阅读算法f2，并回答下列问题：

- (1) 设队列 $Q = (1, 3, 5, 2, 4, 6)$ 。写出执行算法f2后的队列 Q ；
- (2) 简述算法f2的功能。

```

void f2 (Queue *Q) {

    DataType e;

    if (!QueueEmpty (Q)) {

        e=DeQueue (Q);

        f2 (Q);

        EnQueue (Q, e);

    }

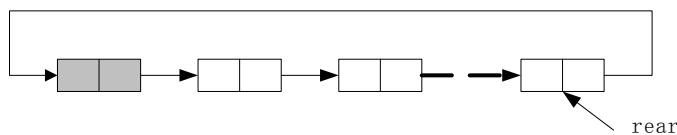
}

```

答案: (1) 6,4,2,5,3,1 (2) 将队列倒置

五、综合题

1、假设以带头结点的循环链表表示队列，并且只设一个指针指向队尾结点，但不设头指针，请写出相应的入队列算法（用函数实现）。



答案: **void EnQueue(Lnode *rear, ElemtType e)**

```

{  Lnode *new;

    New=(Lnode *)malloc(sizeof(Lnode));

    If(!new) return ERROR;

    new->data=e; new->next=rear->next;

    rear->next=new; rear =new;
}

```

}

2、已知 Q 是一个非空队列， S 是一个空栈。编写算法，仅用队列和栈的 ADT 函数和少量工作变量，将队列 Q 的所有元素逆置。

栈的 ADT 函数有：

`void makeEmpty(SqStack s); 置空栈`

`void push(SqStack s, ElemType e); 元素 e 入栈`

`ElemType pop(SqStack s); 出栈，返回栈顶元素`

`int isEmpty(SqStack s); 判断栈空`

队列的 ADT 函数有：

`void enqueue(Queue q, ElemType e); 元素 e 入队`

`ElemType dequeue(Queue q); 出队，返回队头元素`

`int isEmpty(Queue q); 判断队空`

答案： `void QueueInvert(Queue q)`

```
{ ElemType x;  
    makeEmpty(SqStack s);  
    while(!isEmpty(Queue q))  
    {x=deQueue(Queue q);  
     push(SqStack s, ElemTypex);}  
    while(!isEmpty(SqStack s))  
    {x=pop(SqStack s);  
     enQueue(Queue q, ElemType x);}  
}
```

3、对于一个栈，给出输入项 A,B,C,D，如果输入项序列为 A,B,C,D，试给出全部可能的输出序列。

答案：出栈的可能序列：

ABCD ABDC ACDB ACBD ADCB BACD BADC
BCAD BCDA

CBDA CBAD CDBA DCBA

第四章 串

一、选择题

1、设有两个串 s_1 和 s_2 ，求串 s_2 在 s_1 中首次出现位置的运算称作（ C ）。

A. 连接 B. 求子串 C. 模式匹配 D. 判断
子串

2、已知串 $s='aaab'$ ，则 next 数组值为（ A ）。

A. 0123 B. 1123 C. 1231 D.
1211

3、串与普通的线性表相比较，它的特殊性体现在（ C ）。

A. 顺序的存储结构 B. 链式存储结构 C. 数
据元素是一个字符 D. 数据元素任意

4、设串长为 n ，模式串长为 m ，则 KMP 算法所需的附加空间为（ A ）。

A. $O(m)$ B. $O(n)$ C. $O(m*n)$
D. $O(n \log_2 m)$

5、空串和空格串（ B ）。

A. 相同 B. 不相同 C. 可能相同

D. 无法确定

6、与线性表相比，串的插入和删除操作的特点是（ ）。

A. 通常以串整体作为操作对象 B. 需要更多的辅助空间

C. 算法的时间复杂度较高 D. 涉及移动的元素更多

7、设 $SUBSTR(S, i, k)$ 是求 S 中从第 i 个字符开始的连续 k 个字符组成的子串的操作，则对于 $S='Beijing&Nanjing'$ ， $SUBSTR(S, 4, 5)=$ （ B ）。

A. ‘ijing’ B. ‘jing&’ C. ‘ingNa’ D. ‘ing&N’

二、判断题

() 1、造成简单模式匹配算法BF算法执行效率低的原因是有回溯存在。

(√) 2、KMP 算法的最大特点是指示主串的指针不需要回溯。

(√) 3、完全二叉树某结点有右子树，则必然有左子树。

三、填空题

1、求子串在主串中首次出现的位置的运算称为 _____ 模式匹配 _____。

2、设 $s='I\backslash AM\backslash A\backslash TEACHER'$, 其长度是 _____。

3、两个串相等的充分必要条件是两个串的长度相等且 _____ 对应位置
字符相同 _____。

四、程序填空题

1、函数 kmp 实现串的模式匹配，请在空格处将算法补充完整。

```

int kmp(sqstring *s,sqstring *t,int start,int
next[]){
    int i=start-1,j=0;
    while(i<s->len&&j<t->len)
        if(j== -1 || s->data[i]==t->data[j]){
            i++;j++;
        }
        else j= _____;
    if(j>=t->len)
        return( _____ );
    else
        return(-1);
}

```

2、函数实现串的模式匹配算法，请在空格处将算法补充完整。

```

int index_bf(sqstring*s,sqstring *t,int start){
    int i=start-1,j=0;
    while(i<s->len&&j<t->len)
        if(s->data[i]==t->data[j]){
            i++;j++;
        }
        else{
            i= _____ ;j=0;
        }
}

```

```

if(j>=t->len)

    return _____i-t->len+1_____ ;

else

    return -1;

} }/*listDelete*/

```

3、写出下面算法的功能。

```

int function(SqString *s1,SqString *s2) {

    int i;

    for(i=0;i<s1->length&&i<s1->length;i++)

        if(s->data[i]!=s2->data[i])

            return s1->data[i]-s2->data[i];

    return s1->length-s2->length;

}

```

答案: .串比较算法

4、写出算法的功能。

```

int fun(sqstring *s,sqstring *t,int start){

    int i=start-1,j=0;

    while(i<s->len&&j<t->len)

        if(s->data[i]==t->data[j]){

            i++;j++;

        }else{

            i=i-j+1;j=0;
}

```

```

    }

if(j>=t->len)

    return i-t->len+1;

else

    return -1;

}

```

答案：串的模式匹配算法

第五章 数组和广义表

一、选择题

- 1、设广义表L=((a, b, c)), 则L的长度和深度分别为 (C)。
 - A. 1和1
 - B. 1和3
 - C. 1和2
 - D. 2和3

- 2、广义表((a), a)的表尾是 (B)。
 - A. a
 - B. (a)
 - C. ()
 - D. ((a))

- 3、稀疏矩阵的常见压缩存储方法有 (C) 两种。
 - A. 二维数组和三维数组
 - B. 三元组和散列表
 - C. 三元组和十字链表
 - D. 散列表和十字链表

- 4、一个非空广义表的表头 (D)。
 - 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

- 5、数组A[0..5,0..6]的每个元素占5个字节，将其按列优先次序存储在起始地址为1000的内存单元中，则元素A[5][5]的地址是 ()。
 - A. 1175
 - B. 1180
 - C. 1205
 - D. 1210

6、广义表G=(a,b(c,d,(e,f)),g)的长度是 (A)。

- A. 3 B. 4 C. 7 D. 8

7、采用稀疏矩阵的三元组表形式进行压缩存储，若要完成对三元组表进行转置，只要将行和列对换，这种说法 (B)。

- A. 正确 B. 错误 C. 无法确定 D. 以上均不对

8、广义表(a,b,c)的表尾是 (B)。

- A. b,c B. (b,c) C. c D. (c)

9、常对数组进行两种基本操作是 ()。

- A. 建立和删除 B. 索引和修改 C. 查找和修改
D. 查找与索引

10、对一些特殊矩阵采用压缩存储的目的主要是为了 (D)。

- A. 表达变得简单 B. 对矩阵元素的存取变得简单
C. 去掉矩阵中的多余元素 D. 减少不必要的存储空间的开销

11、设有一个10阶的对称矩阵A，采用压缩存储方式，以行序为主存储，
a₁₁为第一个元素，其存储地址为1，每元素占1个地址空间，则a₈₅的地
址为 ()。

- A. 13 B. 33 C. 18 D. 40

12、设矩阵A是一个对称矩阵，为了节省存储，将其下三角部分按行序存
放在一维数组B[1,n(n-1)/2]中，对下三角部分中任一元素

$a_{i,j}$ ($i >= j$)，在一维数组B的下标位置k的值是 (B)。

A. $i(i-1)/2+j-1$

B. $i(i-1)/2+j$

C.

$i(i+1)/2+j-1$

D. $i(i+1)/2+j$

13、广义表A=((a), a) 的表头是 (B)。

A. a

B. (a)

C. b

D. ((a))

14、稀疏矩阵一般的压缩存储方法有两种，即 (C)。

A. 二维数组和三维数组

B. 三元组和散列

C. 三元

组和十字链表

D. 散列和十字链表

15、假设以三元组表表示稀疏矩阵，则与如图所示三元组表对应的 4×5 的稀疏矩阵是（注：矩阵的行列下标均从 1 开始）(B)。

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

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

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

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

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

16、以下有关广义表的表述中，正确的是 ()。

A. 由 0 个或多个原子或子表构成的有限序列

B. 至少

有一个元素是子表

C. 不能递归定义

D. 不能为空表

17、对广义表 L=((a,b), ((c,d), (e,f))) 执行

`head(tail(head(tail(L))))` 操作的结果是 ()。

A. 的

B. e

C. (e)

D. (e,f)

二、判断题

- () 1、广义表中原子个数即为广义表的长度。
- () 2、一个稀疏矩阵采用三元组表示，若把三元组中有关行下标与列下标的值互换，并把 mu 和 nu 的值进行互换，则完成了矩阵转置。
- (√) 3、稀疏矩阵压缩存储后，必会失去随机存取功能。
- () 4、广义表的长度是指广义表中括号嵌套的层数。
- (√) 5、广义表是一种多层次的数据结构，其元素可以是单原子也可以是子表。

三、填空题

1、已知二维数组A[m][n]采用行序为主方式存储，每个元素占k个存储单元，并且第一个元素的存储地址是LOC(A[0][0])，则A[i][j]的地址是____
Loc(A[0][0]) + (i*N+j)*k ____。

2、广义表运算式HEAD(TAIL((a,b,c),(x,y,z)))的结果是：
(x,y,z)。

3、二维数组，可以按照_____两种不同的存储方式。

4、稀疏矩阵的压缩存储方式有：_____ 和 _____。

四、综合题

1、现有一个稀疏矩阵，请给出它的三元组表。

$$\begin{bmatrix} 0 & 3 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & -2 & 0 \end{bmatrix}$$

答案：

i	j	v
1	2	3
1	3	1
2	1	1
3	2	2
3	3	1
4	3	-2

第六章 树

一、选择题

$\sqrt{2^k - 1}$

1、二叉树的深度为 k ，则二叉树最多有 (C) 个结点。

- A. $2k$ B. 2^{k-1} C. ~~$2^k - 1$~~ D. $2k - 1$

2、用顺序存储的方法，将完全二叉树中所有结点按层逐个从左到右的顺序存放在一维数组 $R[1..N]$ 中，若结点 $R[i]$ 有右孩子，则其右孩子是 (B)。

- A. $R[2i-1]$ B. $R[2i+1]$ C. $R[2i]$
D. $R[2/i]$

3、设 a, b 为一棵二叉树上的两个结点，在中序遍历时， a 在 b 前面的条件是 (B)。

- A. a 在 b 的右方
B. a 在 b 的左方
C. a 是 b 的祖先
D. a 是 b 的子孙

4、设一棵二叉树的中序遍历序列：badce，后序遍历序列：bdeca，则二叉树先序遍历序列为 ()。

- A. adbce B. decab C. debac D. abcde

5、在一棵具有 5 层的满二叉树中结点总数为 ()。

$\sqrt{2^5 - 1}$

A. 31 B. 32 C. 33 D. 16

6、由二叉树的前序和后序遍历序列（B）惟一确定这棵二叉树。

A. 能 B. 不能

7、某二叉树的中序序列为ABCDEFG，后序序列为BDCAFGE，则其左子树中结点数目为（C）。

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

8、若以{4,5,6,7,8}作为权值构造哈夫曼树，则该树的带权路径长度为（C）。

A. 67 B. 68 C. 69 D. 70

9、将一棵有100个结点的完全二叉树从根这一层开始，每一层上从左到右依次对结点进行编号，根结点的编号为1，则编号为49的结点的左孩子编号为（A）。

A. 98 B. 99 C. 50 D. 48

10、表达式 $a * (b + c) - d$ 的后缀表达式是（B）。

A. abcd+- B. abc+*d- C. abc*+d- D.
-+*abcd

11、对某二叉树进行先序遍历的结果为ABDEFCA，中序遍历的结果为DBFEAC，则后序遍历的结果是（B）。

A. DBFEAC B. DFEBCA C. BDFECA D. BDEFAC

12、树最适合用来表示（C）。

A. 有序数据元素 B. 无序数据元素 C. 元素之间具有分支层次关系的数据 D. 元素之间无联系的数据

13、表达式 $A * (B+C) / (D-E+F)$ 的后缀表达式是 (C)。

- A. $A * B + C / D - E + F$ B. $AB * C + D / E - F +$ C. $ABC + * DE - F + /$
D. $ABCDED * + / - +$

14、在线索二叉树中，t所指结点没有左子树的充要条件是 ()。

- A. $t \rightarrow left == NULL$ B. $t \rightarrow ltag == 1$ C.
 $t \rightarrow ltag == 1 \& \& t \rightarrow left == NULL$ D. 以上都不对

15、任何一棵二叉树的叶结点在先序、中序和后序遍历序列中的相对次序 ()。

- A. 不发生改变 B. 发生改变 C. 不能确定
D. 以上都不对

16、假定在一棵二叉树中，度为2的结点数为15，度为1的结点数为30，则叶子结点数为 () 个。

- A. 15 B. 16 C. 17 D. 47

17、在下列情况中，可称为二叉树的是 (B)。

- A. 每个结点至多有两棵子树的树 B. 哈夫曼树
C. 每个结点至多有两棵子树的有序树 D. 每个结点只有一棵子树

18、用顺序存储的方法，将完全二叉树中所有结点按层逐个从左到右的顺序存放在一维数组 R[1..n] 中，若结点 R[i] 有左孩子，则其左孩子是 (C)。

- A. $R[2i-1]$ B. $R[2i+1]$ C. $R[2i]$
D. $R[2/i]$

19、下面说法中正确的是 (D)。

- A. 度为2的树是二叉树
是二叉树
- B. 度为2的有序树
- C. 子树有严格左右之分的树是二叉树
- D. 子树有严格左右之分，且度不超过2的树是二叉树

20、树的先根序列等同于与该树对应的二叉树的 ()。

- A. 先序序列
- B. 中序序列
- C. 后序序列
- D. 层序序列

21、按照二叉树的定义，具有3个结点的二叉树有 (C) 种。

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

22、由权值为3, 6, 7, 2, 5的叶子结点生成一棵哈夫曼树，它的带权路径长度为 (A)。

- A. 51 B. 23 C. 53 D. 74

二、判断题

- (√) 1、存在这样的二叉树，对它采用任何次序的遍历，结果相同。 $15+36$
- (×) 2、中序遍历一棵二叉排序树的结点，可得到排好序的结点序列。
- (×) 3、对于任意非空二叉树，要设计其后序遍历的非递归算法而不使用堆栈结构，最适合的方法是对该二叉树采用三叉链表。
- (×) 4、在哈夫曼编码中，当两个字符出现的频率相同时，其编码也相同，对于这种情况应做特殊处理。
- (√) 5、一个含有n个结点的完全二叉树，它的高度是 $\lfloor \log_2 n \rfloor + 1$ 。
- (√) 6、完全二叉树的某结点若无左孩子，则它必是叶结点。

三、填空题

1、具有n个结点的完全二叉树的深度是 $\lfloor \log_2 n \rfloor + 1$ 。

2、哈夫曼树是其树的带权路径长度 最小 的二叉树。

3、在一棵二叉树中，度为0的结点的个数是n₀，度为2的结点的个数为n₂，则有n₀= $n_2 + 1$ 。

4、树内各结点度的 最大值 称为树的度。

四、代码填空题

1、函数**InOrderTraverse(Bitree bt)**实现二叉树的中序遍历，请在空格处将算法补充完整。

```
void InOrderTraverse(Bitree bt) {  
    if (          ) {  
        InOrderTraverse(bt->lchild);  
        printf("%c", bt->data);  
                                  ;  
    }  
}
```

2、函数**depth**实现返回二叉树的高度，请在空格处将算法补充完整。

```
int depth(Bitree *t) {  
    if (t==NULL)  
        return 0;  
    else {  
        h1=depth(t->lchild);  
        
```

```

hr=           depth(t->rchild)                  ;  

if(      hl>hr      )  

    return hl+1;  

else  

    return hr+1;  

}  

}

```

3、写出下面算法的功能。

```

Bitree *function(Bitree *bt) {  

    Bitree *t,*t1,*t2;  

    if(bt==NULL)  

        t=NULL;  

    else{  

        t=(Bitree *)malloc(sizeof(Bitree));  

        t->data=bt->data;  

        t1=function(bt->left);  

        t2=function(bt->right);  

        t->left=t2;  

        t->right=t1;  

    }  

    return(t);  

}

```

答案：交换二叉树结点左右子树的递归算法

4、写出下面算法的功能。

```
void function(Bitree *t) {  
    if (p!=NULL) {  
        function (p->lchild);  
        function (p->rchild);  
        printf ("%d", p->data);  
    }  
}
```

答案：二叉树后序遍历递归算法

五、综合题

1、假设以有序对

c

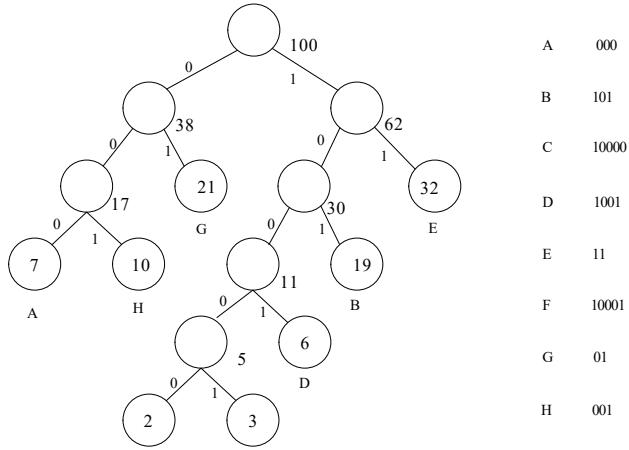
表示从双亲结点到孩子结点的一条边，若已知树中边的集合为

{<a,b>,<a,d>,<a,c>,<c,e>,<c,f>,<c,g>,<c,h>,<e,i>,<e,j>,<g,k>}，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？。

2、假设一棵二叉树的先序序列为 E B A D C F H G I K J，中序序列为 ABCD E F G H I J K，请画出该二叉树。

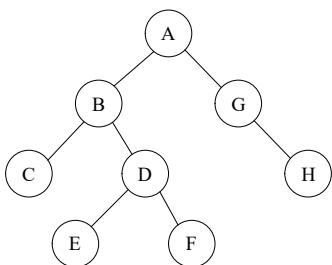
3、假设用于通讯的电文仅由 8 个字母 A、B、C、D、E、F、G、H 组成，字母在电文中出现的频率分别为: 0.07, 0.19, 0.02, 0.06, 0.32, 0.03, 0.21, 0.10。请为这 8 个字母设计哈夫曼编码。



答案：

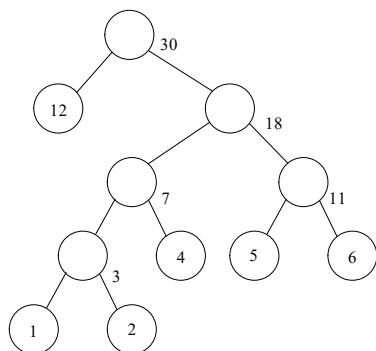
4、已知二叉树的先序遍历序列为 ABCDEFGH，中序遍历序列为 CBEDFAGH，画出二叉树。

答案：二叉树形态



5、试用权集合 {12, 4, 5, 6, 1, 2} 构造哈夫曼树，并计算哈夫曼树的带权路径长度。

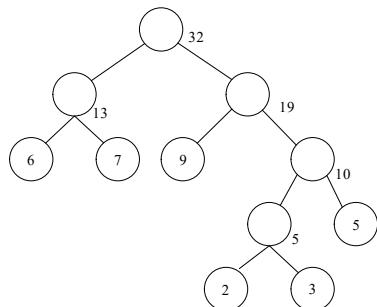
答案：



$$WPL = 12 \cdot 1 + (4+5+6) \cdot 3 + (1+2) \cdot 4 = 12 + 45 + 12 = 69$$

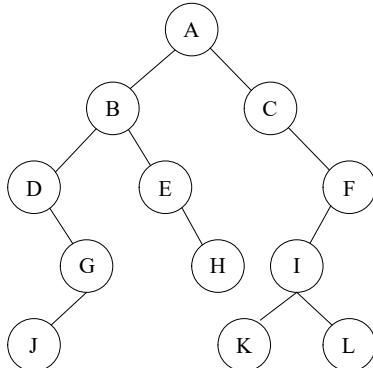
6、已知权值集合为 $\{5, 7, 2, 3, 6, 9\}$ ，要求给出哈夫曼树，并计算带权路径长度 WPL。

答案：(1)树形态：



(2)带权路径长度： $WPL = (6+7+9)*2 + 5*3 + (2+3)*4 = 44 + 15 + 20 = 79$

7、已知一棵二叉树的先序序列：**ABDGJEHCFIKL**； 中序序列：
DJGBEHAACKILF。画出二叉树的形态。



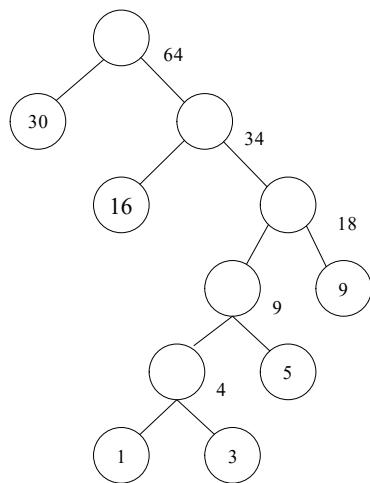
答案：

8、一份电文中有 6 种字符：**A, B, C, D, E, F**，它们的出现频率依次为 16, 5, 9, 3, 30, 1，完成问题：

(1) 设计一棵哈夫曼树；(画出其树结构)

(2) 计算其带权路径长度 WPL；

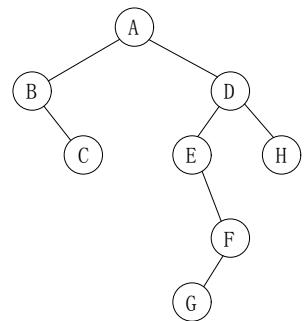
答案：(1)树形态：



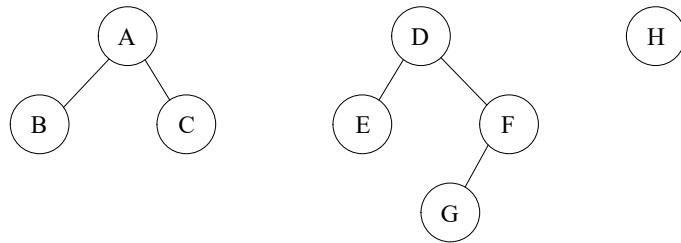
(2) 带 权 路 径 长 度 :

$$WPL = 30*1 + 16*2 + 9*3 + 5*4 + (1+3)*5 = 30 + 32 + 27 + 20 + 20 = 129$$

9、已知某森林的二叉树如下所示，试画出它所表示的森林。

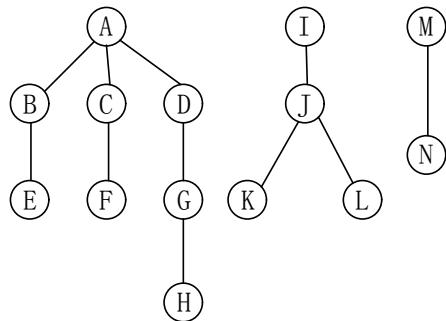


答案：

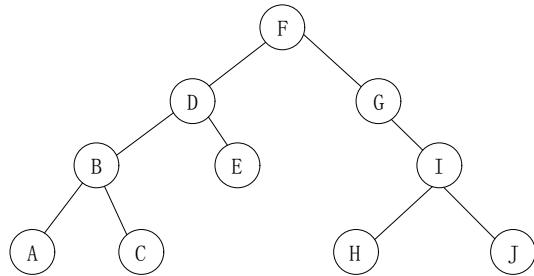


10、有一份电文共使用 5 个字符;a, b, c, d, e, 它们的出现频率依次为 4、7、5、2、9, 试构造哈夫曼树，并给出每个字符的哈夫曼编码。

11、画出与下图所示的森林相对应的二叉树，并指出森林中的叶子结点在二叉树中具有什么特点。



12、如下所示的二叉树，请写出先序、中序、后序遍历的序列。



答案：先序：**FDBACEGIHJ**

中序：**ABCDEFIGHJ**

后序：**ACBEDHJIGF**

六、编程题

1、编写求一棵二叉树中结点总数的算法。

答案：（以先序遍历的方法为例）

```
void count_preorder(Bitree *t, int *n)
{
    if(t!=NULL)
        {*n++;
        count_preorder(t->lchild);
        count_preorder(t->rchild); }
```

}

第七章 图

一、选择题

- 1、12、对于具有n个顶点的图，若采用邻接矩阵表示，则该矩阵的大小为(B)。
- A. n B. n^2 C. $n-1$ D. $(n-1)^2$
- 2、如果从无向图的任一顶点出发进行一次深度优先搜索即可访问所有顶点，则该图一定是（ ）。
- A. 完全图//每对不同的顶点之间都恰连有一条边相连
B. 连通图 C. 有回路 D. 一棵树
- 3、关键路径是事件结点网络中（ ）。
- A. 从源点到汇点的最长路径 B. 从源点到汇点的最短路径
C. 最长的回路 D. 最短的回路
- 4、下面（ ）可以判断出一个有向图中是否有环（回路）。
- A. 广度优先遍历 B. 拓扑排序
C. 求最短路径 D. 求关键路径
- 5、带权有向图G用邻接矩阵A存储，则顶点i的入度等于A中（ ）。

A. 第*i*行非无穷的元素之和 B. 第*i*列非无穷的元素个数之
和

C. 第*i*行非无穷且非0的元素个数 D. 第*i*行与第*i*列非无穷且非
0的元素之和

6、采用邻接表存储的图，其深度优先遍历类似于二叉树的 (B)。

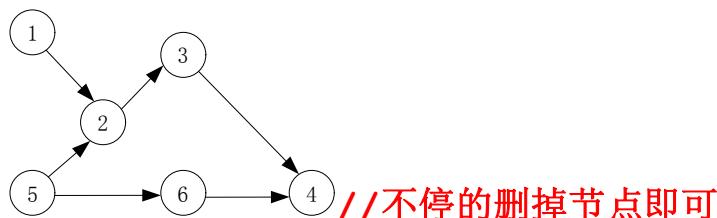
A. 中序遍历 B. 先序遍历 C. 后序遍历 D.
按层次遍历

7、无向图的邻接矩阵是一个 (A)。
A. 对称矩阵 B. 零矩阵 C. 上三角矩阵 D. 对
角矩阵

8、当利用大小为N的数组存储循环队列时，该队列的最大长度是 ()。
A. N-2 B. N-1 C. N D. N+1

9、邻接表是图的一种 ()。
A. 顺序存储结构 B. 链式存储结构 C. 索引存储结构 D.
散列存储结构

10、下面有向图所示的拓扑排序的结果序列是 ()。
A. 125634 B. 516234 C. 123456 D. 521643



11、在无向图中定义顶点 v_i 与 v_j 之间的路径为从 v_i 到 v_j 的一个

()。

A. 顶点序列 B. 边序列 C. 权值总和

D. 边的条数

12、在有向图的逆邻接表中，每个顶点邻接表链接着该顶点所有（ ）邻接点。

A. 入边 B. 出边 C. 入边和出边 D. 不是出边

也不是入边

13、设 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$ 为两个图，如果 $V_1 \subseteq V_2, E_1 \subseteq E_2$ 则称（ ）。

A. G_1 是 G_2 的子图 B. G_2 是 G_1 的子图 C. G_1 是 G_2

的连通分量 D. G_2 是 G_1 的连通分量

14、已知一个有向图的邻接矩阵表示，要删除所有从第*i*个结点发出的边，应（ ）。

A. 将邻接矩阵的第*i*行删除 B. 将邻接矩阵的第*i*行元素全

部置为0 C. 将邻接矩阵的第*i*列删除 D. 将邻接矩阵的第*i*列元素全部置为0

15、任一个有向图的拓扑序列（ ）。

A. 不存在 B. 有一个 C. 一定有多个 D. 有一个或

多个

16、在一个有向图中，所有顶点的入度之和等于所有顶点的出度之和的（ ）倍。

A. $1/2$ B. 1 C. 2 D. 4

17、下列关于图遍历的说法不正确的是（ ）。

- A. 连通图的深度优先搜索是一个递归过程
- B. 图的广度优先搜索中邻接点的寻找具有“先进先出”的特征
- C. 非连通图不能用深度优先搜索法
- D. 图的遍历要求每一顶点仅被访问一次

18、带权有向图G用邻接矩阵A存储，则顶点i的入度为A中：（ ）。

- A. 第i行非∞的元素之和
- B. 第i列非∞的元素之和
- C. 第i行非∞且非0的元素个数
- D. 第i列非∞且非0的元素个数

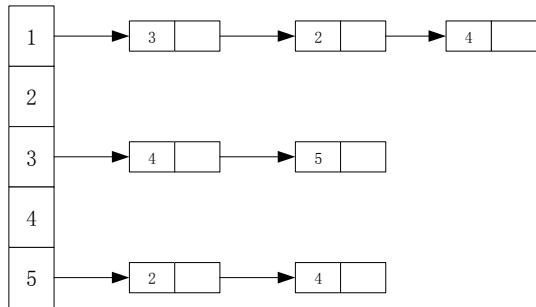
19、采用邻接表存储的图的广度优先遍历算法类似于二叉树的（ ）。

- A. 先序遍历
- B. 中序遍历
- C. 后序遍历
- D. 按层次遍历

20、一个具有n个顶点的有向图最多有（ ）条边。

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

21、已知一个有向图的邻接表存储结构如图所示，根据深度优先遍历算法，从顶点v1出发，所得到的顶点序列是（ ）。



- A. v1,v2,v3,v5,v4 B. v1,v2,v3,v4,v5
 C. v1,v3,v4,v5,v2 D. v1,v4,v3,v5,v2

22、关键路径是事件结点网络中（ ）。

- A. 从源点到汇点的最长路径 B. 从源点到汇点的最短路径
 C. 最长的回路 D. 最短的回路

23、以下说法正确的是 (B)。

- A. 连通分量是无向图中的极小连通子图
 B. 强连通分量是有向图中的极大强连通子图
 C. 在一个有向图的拓扑序列中若顶点a在顶点b之前，则图中必有一条弧<a,b>
 D. 对有向图G，如果以任一顶点出发进行一次深度优先或广度优先搜索能访问到每个顶点，则该图一定是完全图

24、假设有向图含n个顶点及e条弧，则表示该图的邻接表中包含的弧结点个数为 (C)。

- A. n B. e C. 2e D. n*e

25、设图的邻接矩阵为 $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ ，则该图为 (A)。



A. 有向图

B. 无向图

C. 强连通图

D.

完全图

26、为便于判别有向图中是否存在回路，可借助于（ ）。

A. 广度优先搜索算法

B. 最小生成树算法

C. 最短路径算法

D. 拓扑排序算法

27、任何一个无向连通图的最小生成树（ ）种。

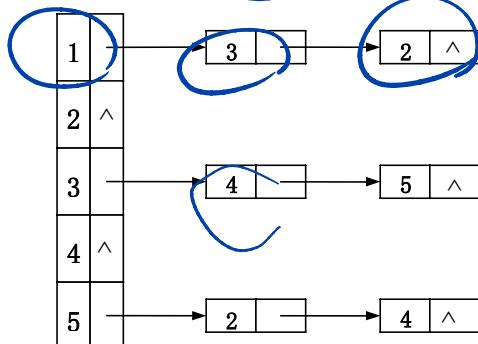
A. 只有一棵

B. 有一棵或多棵

C. 一定有多棵

D. 可能不存在

28、已知一有向图的邻接表存储结构如图所示，根据有向图的广度优先遍历算法，从顶点v1出发，所得到的顶点序列是（ ）。



A. v1,v2,v3,v4,v5

B. v1,v3,v2,v4,v5

C. v1,v2,v3,v5,v4

D. v1,v4,v3,v5,v2

29、对于一个有向图，若一个顶点的入度为 k_1 、出度为 k_2 ，则对应邻接表中该顶点单链表中的结点数为（ ）。

A. k_1

B. k_2

C. k_1+k_2

D. k_1-k_2

30、一个具有8个顶点的有向图中，所有顶点的入度之和与所有顶点的出度之和的差等于（ ）。

A. 16

B. 4

C. 0

D. 2

31、无向图中一个顶点的度是指图中()。

- 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. 与该顶点连通的顶点数
- D. 通过该顶点的回路数

二、填空题

1、 n 个顶点的连通图至少有 $n-1$ 边。

答案: $n-1$ 条

2、一个连通图的生成树是一个 _____，它包含图中所有顶点，但只有足以构成一棵树的 $n-1$ 条边。

答案: 极小连通子图

3、一个图的 _____ 表示法是惟一的。

答案: 邻接矩阵

4、遍历图的基本方法有深度优先搜索和广度优先搜索，其中 _____ 是一个递归过程。

答案: 深度优先搜索

5、在无向图 G 的邻接矩阵 A 中，若 $A[i][j]$ 等于 1，则 $A[j][i]$ 等于 _____。

答案: 1

6、判定一个有向图是否存在回路可以利用 _____。

答案: 拓扑排序

7、已知一个图的邻接矩阵表示，计算第 i 个结点的入度的方法

是 结点。

8、 n 个顶点的无向图最多有 $n-1$ 边。

9、已知一个图的邻接矩阵表示，删除所有从第*i*个结点出发的边的方法是 _____。

10、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 _____。

三、判断题

1、图的连通分量是无向图的极小连通子图。 x

2、一个图的广度优先搜索树是惟一的。 x

3、图的深度优先搜索序列和广度优先搜索序列不是惟一的。 ✓

4、邻接表只能用于存储有向图，而邻接矩阵则可存储有向图和无向图。

x

5、存储图的邻接矩阵中，邻接矩阵的大小不但与图的顶点个数有关，而且与图的边数也有关。 x

6、AOV网是一个带权的有向图。 x

7、从源点到终点的最短路径是唯一的。 x

8、邻接表只能用于存储有向图，而邻接矩阵则可存储有向图和无向图。

x

9、图的生成树是惟一的。 x

四、程序分析题

1、写出下面算法的功能。

```
typedef struct{
```

```

int vexnum,arcnum;

char vexts[N];

int arcs[N][N];

}graph;

void funtion(int i,graph *g){

    int j;

    printf("node:%c\n",g->vexts[i]);

    visited[i]=TRUE;

    for(j=0;j<g->vexnum;j++)

        if((g->arcs[i][j]==1)&&(!visited[j]))

            function(j,g);

}

```

答案：实现图的深度优先遍历算法

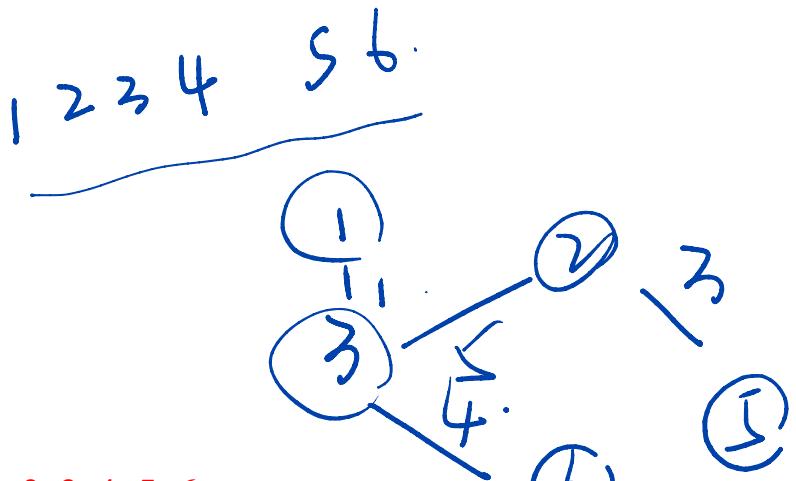


五、综合题

1、已知图 G 的邻接矩阵如下所示：

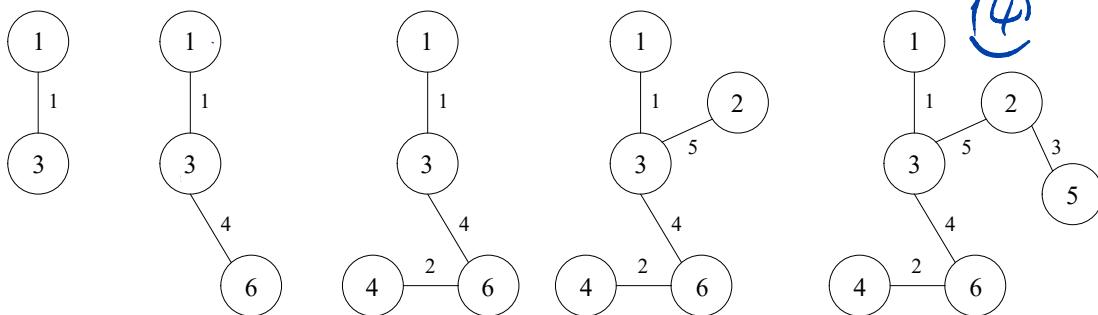
- (1) 求从顶点 1 出发的广度优先搜索序列；
- (2) 根据 prim 算法，求图 G 从顶点 1 出发的最小生成树，要求表示出其每一步生成过程。（用图或者表的方式均可）。

∞	6	1	5	∞	∞
6	∞	5	∞	3	∞
1	5	∞	5	6	4
5	∞	5	∞	∞	2
∞	3	6	∞	∞	6
∞	∞	4	2	6	∞



答案: (1) 广度优先遍历序列: 1; 2, 3, 4; 5; 6

(2) 最小生成树 (prim 算法)



6与2为6

2、设一个无向图的邻接矩阵如下图所示:

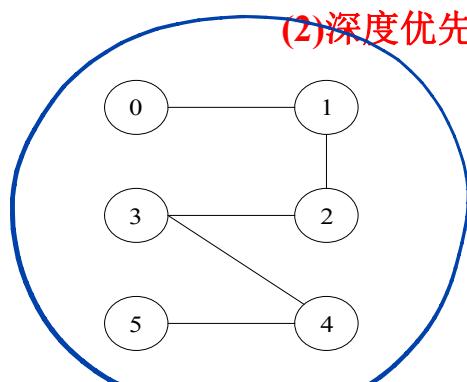
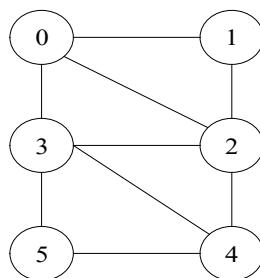
(1) 画出该图;

(2) 画出从顶点 0 出发的深度优先生成树;

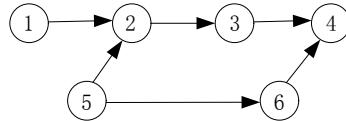
0	1	1	1	0	0
1	0	1	0	0	0
1	1	0	1	1	0
1	0	1	0	1	1
0	0	1	1	0	1
0	0	0	1	1	0

答案: (1) 图形态

(2) 深度优先搜索树



3、写出下图中全部可能的拓扑排序序列。

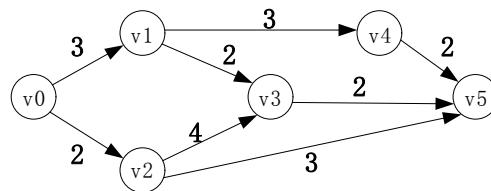


答案: 1, 5, 2, 3, 6, 4 1, 5, 6, 2, 3, 4 5, 1, 2, 3, 6,

4

5, 1, 6, 2, 3, 4 5, 6, 1, 2, 3, 4

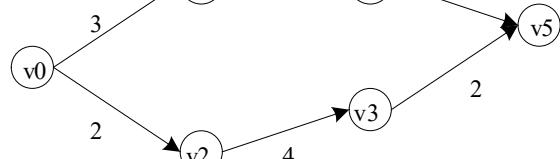
4、AOE网G如下所示，求关键路径。(要求标明每个顶点的最早发生时间和最迟发生时间，并画出关键路径)



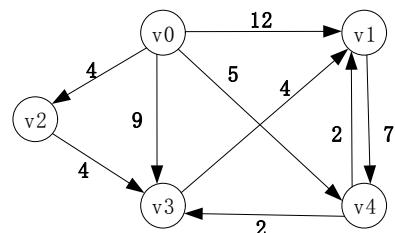
答案: (1)最早发生时间和最迟发生时间:

顶点	ve	vl
v0	0	0
v1	3	3
v2	2	2
v3	6	6
v4	6	6
v5	8	8

(2)关键路径:



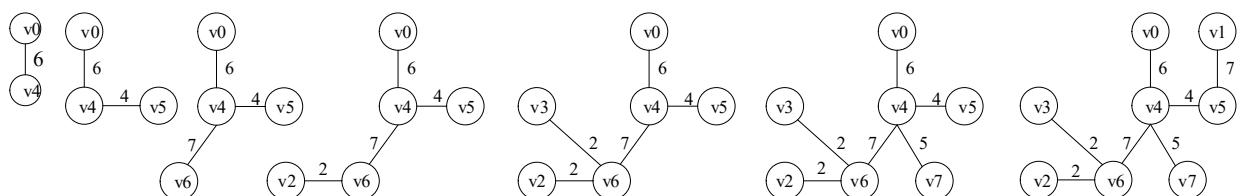
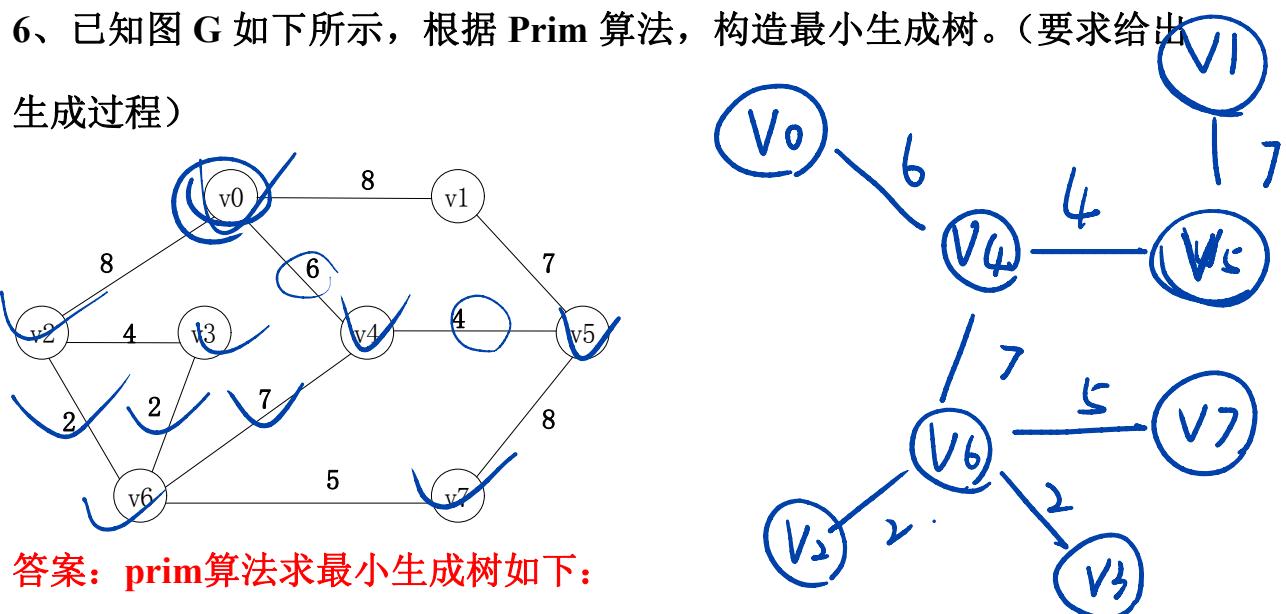
5、已知有向图 G 如下所示，根据迪杰斯特拉算法求顶点 v0 到其他顶点的最短距离。(给出求解过程)



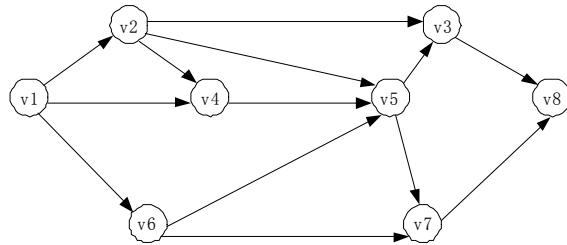
答案:

终点	从 v0 到各终点的 d 值和最短路径的求解过程
----	--------------------------

	i=1	i=2	i=3	i=4
v1	12 (v0,v1)	12 (v0,v1)	7 (v0,v4,v1)	
v2	4 (v0,v2)			
v3	9 (v0,v3)	8 (v0,v2,v3)	7 (v0,v4,v3)	7 (v0,v4,v3)
v4	5 (v0,v4)	5 (v0,v4)		
vj	v2	v4	v1	v3
s	{v0,v2}	{v0,v4}	{v0,v4,v1}	{v0,v4,v3}



7、已知有向图如下所示，请写出该图所有的拓扑序列。



答案：拓扑排序如下：

v1, v2, v4, v6, v5, v3, v7, v8

v1, v2, v4, v6, v5, v7, v3, v8

v1, v2, v6, v4, v5, v3, v7, v8

v1, v2, v6, v4, v5, v7, v3, v8

v1, v6, v2, v4, v5, v3, v7, v8

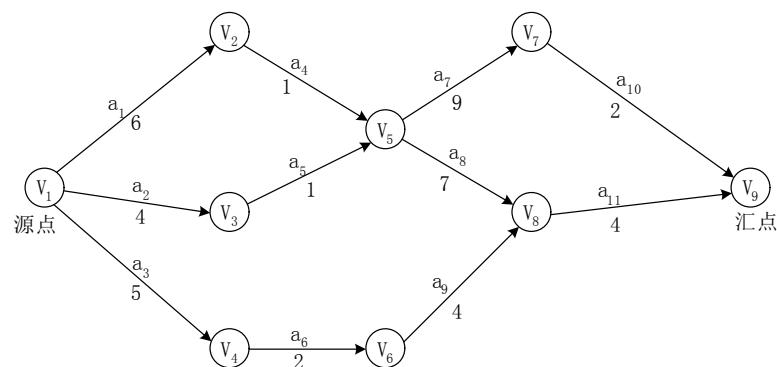
v1, v6, v2, v4, v5, v7, v3, v8

8、如下图所示的 AOE 网，求：

(1) 求事件的最早开始时间 ve 和最迟开始时间 vl；

事件	1	2	3	4	5	6	7	8	9
ve									
vl									

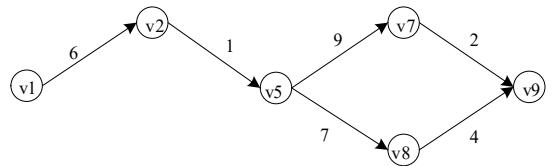
(2) 求出关键路径；



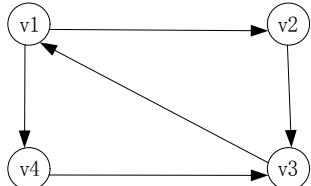
答案：(1)求 ve 和 vl

(2)关键路径

事件	1	2	3	4	5	6	7	8	9
ve	0	6	4	5	7	7	16	14	18
vl	0	6	6	8	7	10	16	14	18
*	*	*			*	*	*	*	*



如下所示的有向图，回答下面问题：



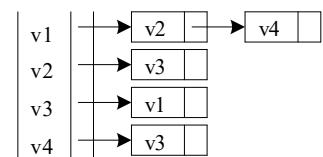
(1) 该图是强连通的吗？若不是，给出强连通分量。

(2) 请给出图的邻接矩阵和邻接表表示。

答案：(1) 是强连通图

(2) 邻接矩阵和邻接表为：

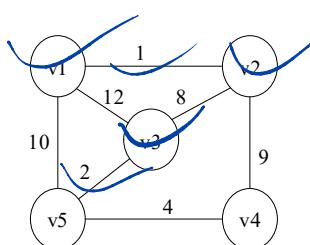
$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



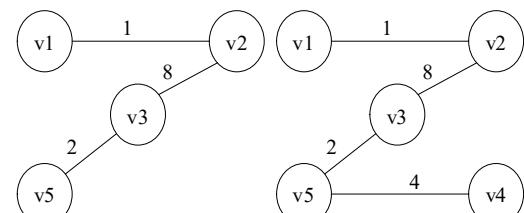
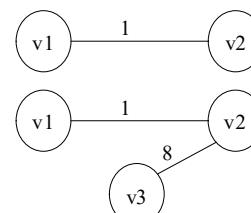
9、已知图 G 的邻接矩阵 $A = \begin{bmatrix} \infty & 1 & 12 & 6 & 10 \\ 1 & \infty & 8 & 9 & \infty \\ 12 & 8 & \infty & \infty & 2 \\ 6 & 9 & \infty & \infty & 4 \\ 10 & \infty & 2 & 4 & \infty \end{bmatrix}$ ，试画出它所表示的图 G，并根据 Prim 算法求出图的最小生成树（给出生成过程）。

答案：

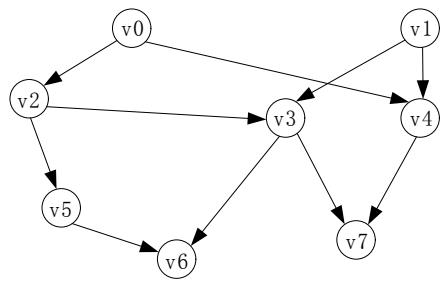
(1) 图形态：



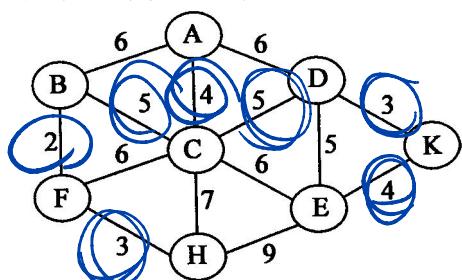
(2) prim 算法求最小生成树：



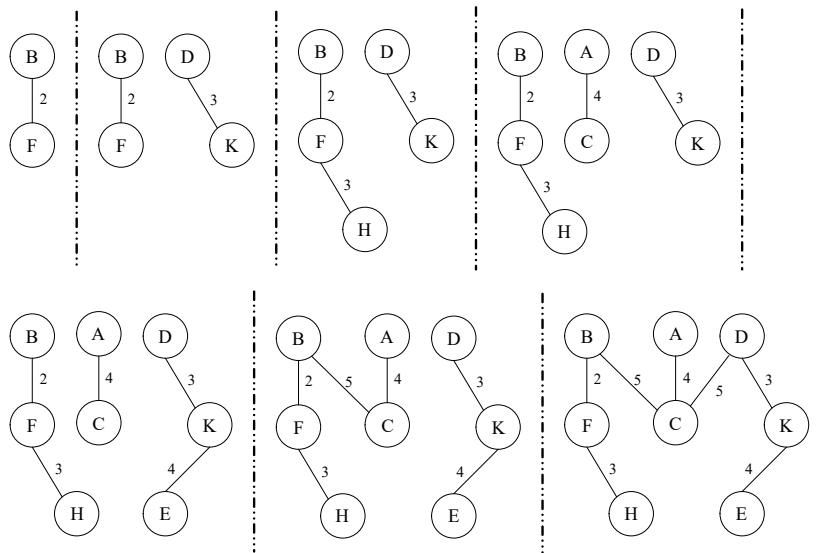
10、如下图所示的 AOV 网，写出其中三种拓扑排序序列。



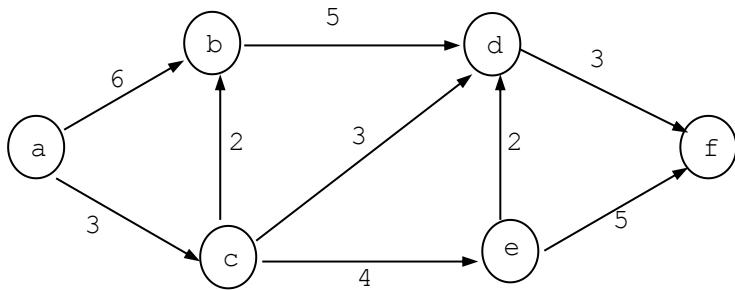
11、已知图 G 如下，根据克鲁斯卡尔算法求图 G 的一棵最小生成树。（要求给出构造过程）



答案：kruskal算法的最小生成树



12、已知图 G 如下所示，求从顶点 a 到其余各顶点的最短路径。（给出求解过程）



答案：

终点	最短路径求解过程				
b	6 (a,b)	5 (a,c,b)			
c	3 (a,c)				
d	∞	6 (a,c,d)	6 (a,c,d)		
e	∞	7 (a,c,e)	7 (a,c,e)	7 (a,c,e)	
f	∞	∞	∞	9 (a,c,d,f)	9 (a,c,d,f)
vj	c	b	d	e	f
S	{a,c}	{a,c,b}	{a,c,d}	{a,c,e}	{a,c,d,f}

第九章 查找

一、选择题

1、已知一个有序表为 (11, 22, 33, 44, 55, 66, 77, 88, 99)，则

折半查找55需要比较（ A ）次。

- A. 1 B. 2 C. 3 D. 4

2、有一组关键字序列{13,16,6,34,32,98,73,1,27}，哈希表的表长为13，哈希函数为 $H(key) = key \bmod 13$ ，冲突解决的办法为链地址法，请构造哈希表（用图表示）。

3、解决哈希冲突的主要方法有（ ）。

- A. 数字分析法、除余法、平方取中法 B. 数字分析法、除余法、线性探测法
C. 数字分析法、线性探测法、再哈希法 D. 线性探测法、再哈希法、链地址法

4、在一棵深度为h的具有n个元素的二叉排序树中，查找所有元素的最长查找长度为（ ）。

- A. n B. $\log_2 n$ C. $(h+1)/2$ D. h

5、已知表长为25的哈希表，用除留取余法，按公式 $H(key) = key \bmod p$ 建立哈希表，则p应取（ ）为宜。

- A. 23 B. 24 C. 25 D. 26

6、设哈希表长m=14，哈希函数 $H(key) = key \bmod 11$ 。表中已有4个结点： $addr(15)=4, addr(38)=5, addr(61)=6, addr(84)=7$ 其余地址为空，如用二次探测再散列处理冲突，则关键字为49的地址为（ A ）。

- A. 8 B. 3 C. 5 D. 9

7、在散列查找中，平均查找长度主要与（ c ）有关。

- A. 散列表长度 B. 散列元素个数 C. 装填因子

D. 处理冲突方法

8、根据一组记录（56，42，50，64，48）依次插入结点生成一棵AVL树，当插入到值为_____的结点时需要进行旋转调整。

9、m阶B-树中的m是指（ ）。

- A. 每个结点至少具有m棵子树
- B. 每个结点最多具有m棵子树
- C. 分支结点中包含的关键字的个数
- D. m阶B-树的深度

10、一个待散列的线性表为 $k=\{18, 25, 63, 50, 42, 32, 9\}$ ，散列函数为 $H(k)=k \bmod 9$ ，与18发生冲突的元素有（ ）个。

- A. 1
- B. 2
- C. 3
- D. 4

11、在对查找表的查找过程中，若被查找的数据元素不存在，则把该数据元素插到集合中，这种方式主要适合于（ ）。

- A. 静态查找表
- B. 动态查找表
- C. 静态查找表和动态查找表
- D. 两种表都不适合

12、有一个有序表为 $\{1, 3, 9, 12, 32, 41, 45, 62, 75, 77, 82, 95, 100\}$ ，当折半查找值为82的结点时，（ B ）次比较后查找成功。

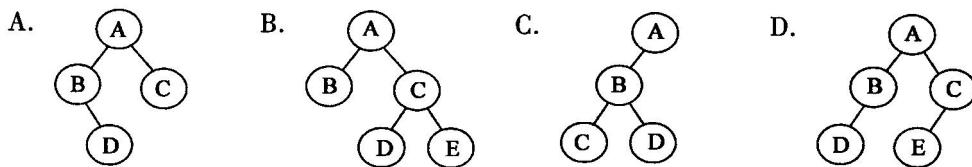
- A. 1
- B. 4
- C. 2
- D. 8

13、在各种查找方法中，平均查找承担与结点个数n无关的查找方法是（ C ）。

- A. 顺序查找
- B. 折半查找
- C. 哈希查找

D. 分块查找

14、下列二叉树中，不平衡的二叉树是（ C ）。



15、对一棵二叉排序树按（ B ）遍历，可得到结点值从小到大的排列序列。

- A. 先序 B. 中序 C. 后序 D. 层次

16、解决散列法中出现的冲突问题常采用的方法是（ D ）。

- A. 数字分析法、除余法、平方取中法 B. 数字分析法、除余法、线性探测法
C. 数字分析法、线性探测法、多重散列法 D. 线性探测法、多重散列法、链地址法

17、对线性表进行折半查找时，要求线性表必须（ C ）。

- A. 以顺序方式存储 B. 以链接方式存储
C. 以顺序方式存储，且结点按关键字有序排序 D. 以链接方式存储，且结点按关键字有序排序

二、填空题

1、在散列函数 $H(key) = key \% p$ 中， p 应取 _____。

2、已知有序表为（12，18，24，35，47，50，62，83，90，115，134），当用折半查找90时，需进行 _____ 次查找可确定成功。

3、具有相同函数值的关键字对哈希函数来说称为 _____。

4、在一棵二叉排序树上实施_____遍历后，其关键字序列是一个有序表。

5、在散列存储中，装填因子 α 的值越大，则存取元素时发生冲突的可能性就越_____大_____； α 值越小，则存取元素发生冲突的可能性就越_____小_____。

三、判断题

(\times) 1、折半查找只适用于有序表，包括有序的顺序表和链表。

($\text{ } \text{ }$) 2、二叉排序树的任意一棵子树中，关键字最小的结点必无左孩子，关键字最大的结点必无右孩子。

($\text{ } \text{ }$) 3、哈希表的查找效率主要取决于哈希表造表时所选取的哈希函数和处理冲突的方法。

($\text{ } \text{ }$) 4、平衡二叉树是指左右子树的高度差的绝对值不大于 1 的二叉树。

(\checkmark) 5、AVL 是一棵二叉树，其树上任一结点的平衡因子的绝对值不大于 1。

四、综合题

1、选取哈希函数 $H(k) = (k) \bmod 11$ 。用二次探测再散列处理冲突，试在 0~10 的散列地址空间中对关键字序列 (22, 41, 53, 46, 30, 13, 01, 67) 造哈希表，并求等概率情况下查找成功时的平均查找长度。

答案：(1) 表形态：

0	1	2	3	4	5	6	7	8	9	10
22	01	46	13				30	41	53	
1	1	1	2				3	1	1	

$$(2) ASL: ASL(7) = (1*5 + 2*1 + 3*1)/7 = (5+2+3)/7 = 10/7$$

2、设哈希表 HT 表长 m 为 13，哈希函数为 $H(k) = k \bmod m$ ，给定的关键值序列为 {19, 14, 23, 10, 68, 20, 84, 27, 55, 11}。试求出用线性探

测法解决冲突时所构造的哈希表，并求出在等概率的情况下查找成功的平均查找长度 **ASL**。

答案：(1)表形态：

0	1	2	3	4	5	6	7	8	9	10	11	12	
	14	27	68	55		19	20	84		23	10	11	
1	2	1	2		1	1	3		1	2	2	2	

(2) 平均查找长度： $ASL(10)=(1*5+2*4+3*1)/10=1.6$

3、设散列表容量为 7（散列地址空间 0..6），给定表（30，36，47，52，34），散列函数 $H(K) = K \bmod 6$ ，采用线性探测法解决冲突，要求：

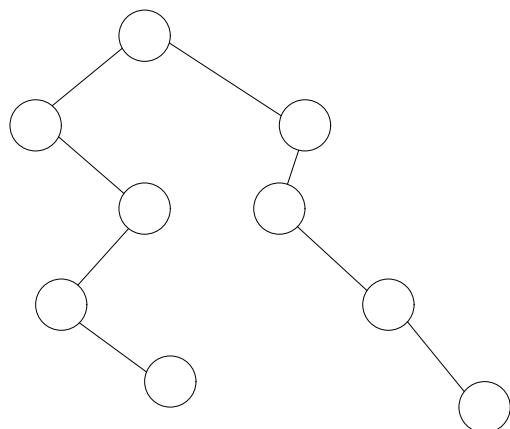
- (1) 构造散列表；
- (2) 求查找数 34 需要比较的次数。

答案：(1)表形态：

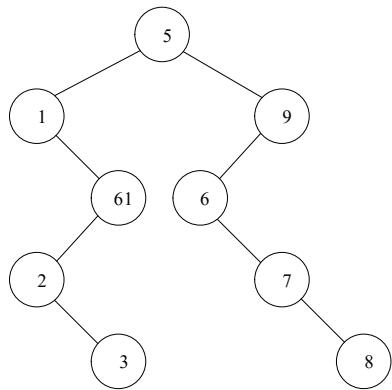
0	1	2	3	4	5	6
30	26			52	47	34
1	2			1	1	3

(2) 查找 34 的比较次数：3

4、已知下面二叉排序树的各结点的值依次为 1—9，请标出各结点的值。



答案：



5、若依次输入序列 {62, 68, 30, 61, 25, 14, 53, 47, 90, 84} 中的元素，生成一棵二叉排序树。画出生成后的二叉排序树（不需画出生成过程）。

6、设有一组关键字 {19, 1, 23, 14, 55, 20, 84, 27, 68, 11, 10, 77}，采用哈希函数 $H(key) = key \bmod 13$ ，采用开放地址法的二次探测再散列方法解决冲突，试在 0—18 的散列空间中对关键字序列构造哈希表，画出哈希表，并求其查找成功时的平均查找长度。

7、已知关键字序列 {11, 2, 13, 26, 5, 18, 4, 9}，设哈希表表长为 16，哈希函数 $H(key) = key \bmod 13$ ，处理冲突的方法为线性探测法，请给出哈希表，并计算在等概率的条件下的平均查找长度。

8、设散列表的长度为 $m=13$ ，散列函数为 $H(k) = k \bmod m$ ，给定的关键码序列为 19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 13, 7，试写出用线性探查法解决冲突时所构造的散列表。

答案：表形态：

0	1	2	3	4	5	6	7	8	9	10	11	12	
13	14	1	68	27	55	19	20	84	7	23	11	1	
1	1	2	1	4	3	1	1	3	3	1	1	1	

9、依次读入给定的整数序列 {7, 16, 4, 8, 20, 9, 6, 18, 5}，构造一棵二

叉排序树，并计算在等概率情况下该二叉排序树的平均查找长度 ASL。

(要求给出构造过程)

10、设有一组关键字 (19, 1, 23, 14, 55, 20, 84, 27, 68, 11, 10, 77)，采用哈希函数 $H(key) = key \% 13$ ，采用二次探测再散列的方法解决冲突，试在 0-18 的散列地址空间中对该关键字序列构造哈希表。

答案：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
27	1	14	55	68	84	19	20		10	23	11	77						
3	1	2	1	2	3	1	1		3	1	1	1						

第十章 内部排序

一、选择题

1、若需要在 $O(n \log_2 n)$ 的时间内完成对数组的排序，且要求排序是稳定的，则可选择的排序方法是（ ）。

- A. 快速排序 B. 堆排序 C. 归并排序 D. 直接插入排序

2、下列排序方法中（ ）方法是不稳定的。

- A. 冒泡排序 B. 选择排序 C. 堆排序 D. 直接插入排序

3、一个序列中有 10000 个元素，若只想得到其中前 10 个最小元素，则最好采用（ ）方法。

- A. 快速排序 B. 堆排序 C. 插入排序 D. 归并

排序

4、一组待排序序列为 (46, 79, 56, 38, 40, 84)，则利用堆排序的方法建立的初始堆为 ()。

A. 79, 46, 56, 38, 40, 80

B.

84, 79, 56, 38, 40, 46

C. 84, 79, 56, 46, 40, 38

D. 84, 56, 79, 40, 46, 38

5、快速排序方法在 () 情况下最不利于发挥其长处。

A. 要排序的数据量太大

B. 要排序的数据中有多个相

同值

C. 要排序的数据已基本有序

D. 要排序的数据个数为奇数

6、排序时扫描待排序记录序列，顺次比较相邻的两个元素的大小，逆序时就交换位置，这是 () 排序的基本思想。

A. 堆排序

B. 直接插入排序

C. 快速排序

D.

冒泡排序

7、在任何情况下，时间复杂度均为 $O(n \log n)$ 的不稳定的排序方法是 ()。

A. 直接插入

B. 快速排序

C. 堆排序

D. 归并

排序

8、如果将所有中国人按照生日来排序，则使用 () 算法最快。

A. 归并排序

B. 希尔排序

C. 快速排序

D. 基

数排序

9、在对 n 个元素的序列进行排序时，堆排序所需要的附加存储空间是

()。

- A. $O(\log_2 n)$ B. $O(1)$ C. $O(n)$ D. $O(n \log_2 n)$

10、排序方法中，从未排序序列中依次取出元素与已排序序列中的元素进行比较，将其放入已排序序列的正确位置上的方法，称为（ ）。

- A. 希尔排序 B. 冒泡排序 C. 插入排序
D. 选择排序

11、一组记录的的序列未（46, 79, 56, 38, 40, 84），则利用堆排序的方法建立的初始堆为（ ）。

- A. 79, 46, 56, 38, 40, 80 B. 84, 79, 56, 38,
40, 46
C. 84, 79, 56, 46, 40, 38 D. 84, 56, 79, 40,
46, 38

12、用某种排序方法对线性表（ 25, 84, 21, 47, 15, 27, 68, 35, 20）进行排序时，元素序列的变化情况如下：

- (1) 25, 84, 21, 47, 15, 27, 68, 35, 20
(2) 20, 15, 21, 25, 47, 27, 68, 35, 84
(3) 15, 20, 21, 25, 35, 27, 47, 68, 84
(4) 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

13、设有1024个无序的元素，希望用最快的速度挑选出其中前5个最大的元素，最好选用（ ）。

- A. 冒泡排序 B. 选择排序 C. 快速排序 D. 堆排序

14、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（ ）。

- A. 快速排序 B. 堆排序 C. 归并排序 D.
基数排序

15、希尔排序的增量序列必须是（ ）。

- A. 递增的 B. 递减的 C. 随机的 D. 非递减的

二、填空题

1、在插入和选择排序中，若初始数据基本正序，则选用_____，若初始数据基本反序，则选用_____。

答案：递增排列 递减排列

2、在插入排序、希尔排序、选择排序、快速排序、堆排序、归并排序和基数排序中，排序是不稳定的有_____。

三、判断题

- 1、直接选择排序是一种稳定的排序方法。 ✓
- 2、快速排序在所有排序方法中最快，而且所需附加空间也最少。 ✗
- 3、直接插入排序是不稳定的排序方法。 ✗
- 4、选择排序是一种不稳定的排序方法。

四、程序分析题

五、综合题

1 、写出用直接插入排序将关键字序列
 $\{54, 23, 89, 48, 64, 50, 25, 90, 34\}$ 排序过程的每一趟结果。

答案：初始： 54, 23, 89, 48, 64, 50, 25, 90, 34

1: (23, 54), 89, 48, 64, 50, 25, 90, 34

2: (23, 54, 89), 48, 64, 50, 25, 90, 34

3: (23, 48, 54, 89), 64, 50, 25, 90, 34

4: (23, 48, 54, 64, 89), 50, 25, 90, 34

5: (23, 48, 50, 54, 64, 89), 25, 90, 34

6: (23, 25, 48, 50, 54, 64, 89), 90, 34

7: (23, 25, 48, 50, 54, 64, 89, 90), 34

8: (23, 25, 48, 50, 54, 64, 89, 90, 34)

2、设待排序序列为 $\{10, 18, 4, 3, 6, 12, 1, 9, 15, 8\}$ 请写出希尔排序每一趟的结果。增量序列为 5, 3, 2, 1。

答案：初始： 10, 18, 4, 3, 6, 12, 1, 9, 15, 8

d=5: 10, 1, 4, 3, 6, 12, 18, 9, 15, 8

d=3: 3, 1, 4, 8, 6, 12, 10, 9, 15, 18

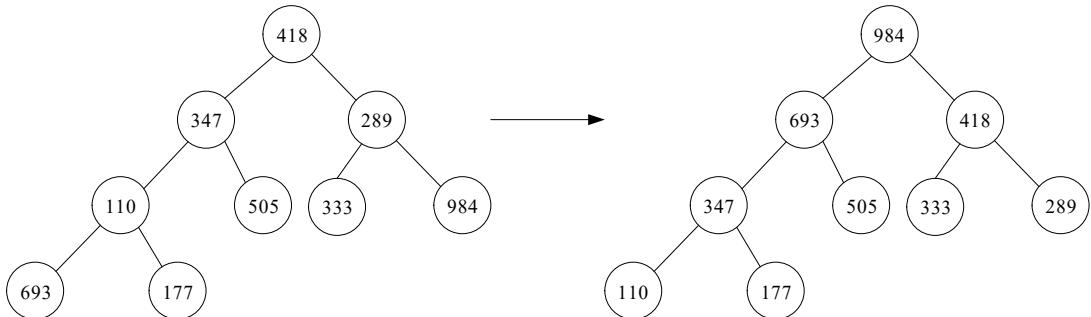
d=2: 3, 1, 4, 8, 6, 9, 10, 12, 15, 18

d=1: 1, 3, 4, 6, 8, 9, 10, 12, 15, 18

3、已知关键字序列 $\{418, 347, 289, 110, 505, 333, 984, 693,$

177}, 按递增排序, 求初始堆(画出初始堆的状态)。

答案: 418, 347, 289, 110, 505, 333, 984, 693, 177



4、有一关键字序列 (265, 301, 751, 129, 937, 863, 742, 694, 076, 438), 写出希尔排序的每趟排序结果。(取增量为 5, 3, 1)

答案：

初始: 265, 301, 751, 129, 937, 863, 742, 694, 076, 438

d=5: 265, 301, 694, 076, 438, 863, 742, 751, 129, 937

d=3: 076, 301, 129, 265, 438, 694, 742, 751, 863, 937

d=1: 076, 129, 265, 301, 438, 694, 742, 751, 863, 937

5、对于直接插入排序，希尔排序，冒泡排序，快速排序，直接选择排序，堆排序和归并排序等排序方法，分别写出：

- (1) 平均时间复杂度低于 $O(n^2)$ 的排序方法;
 - (2) 所需辅助空间最多的排序方法;

答案: (1) 希尔、快速、堆、归并 (2) 归并

6、对关键子序列（72， 87， 61， 23， 94， 16， 05， 58）进行堆排序，使之按关键字递减次序排列（最小堆），请写出排序过程中得到的初始堆和前三趟的序列状态。

答案：

