计算机类-数据结构与算法分析练习题

1. 第一章

1.1 基本概念题

1．线性结构中数据元素的位置之间存在（　　）的关系。 【B】

A．一对多　　B．一对一

C．多对多　　D．每一个元素都有一个直接前驱和一个直接后继

注：D 选不全的，第一个元素就木有前驱，最后一个元素就木有后继。

2．数据结构中，与所使用的计算机无关的是数据的（ ）结构。 【D】

A．物理　　B．存储　　C．逻辑与物理　　D．逻辑

3．结构中的数据元素存在一对一的关系称为_____结构。 【线性】

4．数据元素是数据的基本单位，它（　　）。 【C】

A．只能有一个数据项组成　　B．至少有二个数据项组成

C．可以是一个数据项也可以由若干个数据项组成

D．至少有一个数据项为指针类型

5． 一种逻辑结构（　　）存储结构。 【A】

A．可以有不同的　　B．只能有唯一的

C．的数据元素在计算机中的表示称为　　D．的数据元素之间的关系称为

1.2 逻辑结构题

1．通常数据的逻辑结构包括_____、_____、_____、_____四种类型。

【集合；线性；树形；图状】

2．通常可以把一本含有不同章节的书的目录结构抽象成_____结构。【树形】

3．通常可以把某城市中各公交站点间的线路图抽象成_____结构。【图状】

4．结构中的数据元素存在多对多的关系称为_____结构。【图状（网状）】

5．结构中的数据元素存在一对多的关系称为_____结构。【树形】

6．结构中的数据元素存在一对一的关系称为_____结构。【线性】

1.3 物理结构题

1．把数据存储到计算机中,并具体体现数据之间的逻辑结构称称为物理（ ）结构。【存储】

2．把数据存储到计算机中,并具体体现数据之间的逻辑结构称为_____结构。（物理（存储））

1.4 算法特性题

1．以下特征中，（ ）不是算法的特性。 【C】

A．有穷性 B．确定性 C．有 0 个或多个输出 D．可行性

2．算法的时间复杂度与（ ）有关。 【D】

A．所使用的计算机 B．与计算机的操作系统

C．与数据结构 D．与算法本身

3．要求在 n 个数据元素中找其中值最大的元素，设基本操作为元素间的比较。则比较的次数和算法的时间复杂度分别为_____和 O(n)。 【n-1】

2. 线性表

2.1 基本概念题

1．针对线性表，在存储后如果最常用的操作是取第 i 个结点及其前驱，则采用（ ）存储方式最节省时间。 【B】

A．单链表 B．顺序表 C．单循环链表 D．双链表

2.2 顺序表

1．设有一个长度为 n 的顺序表，要在第 i 个元素之前（也就是插入元素作为新表的第 i 个元素），则移动元素个数为（ ）。 【A】

A．n-i+1 B．n-i C．n-i-1 D．i

注：元素要插入到 i 的位置，不动的是前面 i-1 个元素，总元素有 n 个，要移动的元素个数是：n-(i-1) 即： n- i +1

2. 设顺序存储的线性表长度为 n，对于插入操作，设插入位置是等概率的，则插入一个元素平均移动元素的次数为（ ）。【A】

A. n/2 B. n C. n-1 D. n-i+1

3.设顺序存储的线性表长度为 n，对于删除操作，设删除位置是等概率的，则删除一个元素平均移动元素的次数为（ ）。 【A】

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

4. 线性表的顺序结构中，（ ）。 【C】

A. 逻辑上相邻的元素在物理位置上不一定相邻

B. 数据元素是不能随机访问的

C. 逻辑上相邻的元素在物理位置上也相邻

D. 进行数据元素的插入、删除效率较高

2.3 链表概念题

1. 链表所具备的特点是（ ）。 【D】

A. 可以随机访问任一结点

B. 占用连续的存储空间

C. 可以通过下标对链表进行直接访问

D. 插入删除元素的操作不需要移动元素结点

2. 线性表采用链式存储时，其地址（ ）。 【C】

A. 一定是不连续的 B. 必须是连续的

C. 可以连续也可以不连续 D. 部分地址必须是连续的

3. 设有一个不带头结点的单向循环链表，结点的指针域为 next，指针 p 指向尾结点，现要使 p 指向第一个结点，可用语句_____。 【p=p->next;】

4. 在双向链表中，每个结点有两个指针域，一个指向_____，另一个指向_____。

【结点的直接后继 结点的直接前驱】

5. 以下说法中不正确的是（ ）。 【B】

A. 双向循环链表中每个结点需要包含两个指针域

B. 已知单向链表中任一结点的指针就能访问到链表中每个结点

C. 顺序存储的线性链表是可以随机访问的

D. 单向循环链表中尾结点的指针域中存放的是头指针

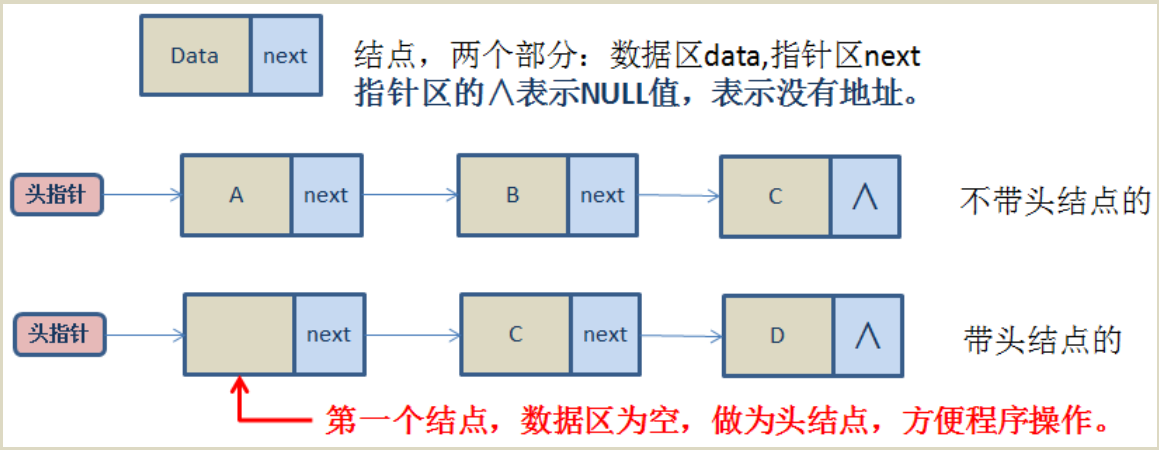
6. 以下表中可以随机访问的是（ ）。 【D】

A. 单向链表 B. 双向链表

C. 单向循环链表 D. 顺序表

2.4 链表指针题

基本图形：



1．在一个单链表中，p、q 分别指向表中两个相邻的结点，且 q 所指结点是 p 所指结点的直接后继，现要删除 q 所指结点，可用的语句是（　　）。 【C】

A．p=q->next　　B．p->next=q　　C．p->next=q→next　D．q->next=NULL

2．在双向链表中，每个结点有两个指针域，一个指向结点的直接后继，另一个指向____。

【结点的直接前驱】

3．设有一个头指针为 head 的单向循环链表，p 指向链表中的结点，若 p->next==_____，则 p 所指结点为尾结点。 【 head 】

4．设有一个头指针为 head 的单向链表，p 指向表中某一个结点，且有 p->next==NULL,通过操作_____，就可使该单向链表构造成单向循环链表。【p->next=head;】

5．带头结点的单向链表的头指针为 head，该链表为空的判定条件是（　　）的值为真。【C】

A．head = = NULL　　B．head->next= =head

C．head->next= = NULL　　D．head = =head->next

6．双向循环链表结点的数据类型为： 【B】

```
struct node
{    int  data;
     struct node *next;    /*指向直接后继*/
     struct node *prior;
 };
```

设 p 指向表中某一结点，要显示 p 所指结点的直接前驱结点的数据元素，可用操作（　　）。

A．printf(“%d”,p->next->data);　　B．printf(“%d”,p->prior->data);

C．printf(“%d”,p->prior->next);　　D．printf(“%d”,p->data);

7．要在一个带头结点的单向循环链表中删除头结点，得到一个新的不带头结点的单向循环链表，若结点的指针域为 next，头指针为 head，尾指针为 p，则可执行 head=head-> next; _____。

【p->next=head;】

8．设有一个头指针为 head 的单向链表，p 指向表中某一个结点，且有 p->next= =NULL,通过操作_____，就可使该单向链表构造成单向循环链表。 【p->next=head;】

9．双向循环链表中，p 指向表中某结点，则通过 p 可以访问到 p 所指结点的直接后继结点和直接前驱结点，这种说法是_____的（回答正确或不正确）。 【正确】

10．设有一个单向链表，结点的指针域为 next，头指针为 head，p 指向尾结点，为了使该单向链表改为单向循环链表，可用语句_____。 【p->next=head;】

11．设有一个单向循环链表，结点的指针域为 next，头指针为 head，指针 p 指向表中某结点，若逻辑表达式_____的结果为真，则 p 所指结点为尾结点。 【p->next= =head;】

12．设有一个单向循环链表，头指针为 head，链表中结点的指针域为 next，p 指向尾结点的直接前驱结点，若要删除尾结点，得到一个新的单向循环链表，可执行操作_____。 【p->next=head;】

13．要在一个单向链表中 p 所指向的结点之后插入一个 s 所指向的新结点，若链表中结点的指针域为 next，可执行_____和 p->next=s;的操作。【s->next= p->next;】

14．要在一个单向链表中删除 p 所指向的结点，已知 q 指向 p 所指结点的直接前驱结点，若链表中结点的指针域为 next，则可执行_____。【q->next= p->next;】

2.5 链表编程题

1．**编程题** 以下是用头插法建立带头结点且有 n 个结点的单向链表的程序，要求结点中的数据域从前向后依次为 n,n-1,……,1，完成程序中空格部分。

```
NODE *create2(n)
{    NODE *head , *p, *q;
      int   i;
      p=(NODE*)malloc(sizeof(NODE));
      p->next=NULL;
      head=（1）          ;
           （2）          ;
      for(i=1; i<=n; i++)
      { p=  （3）          ;
        p->data=i;
        if(i= =1)
            p->next=NULL;
        else
            p->next=  （4）          ;
        q->next=  （5）          ;
        }
        return(head);
}
```

下面答案： （1）p

（2）q=p

（3）(NODE*)malloc(sizeof(NODE))
（4）q->next
（5）p

2．**编程题** 以下函数在 head 为头指针的具有头结点的单向链表中删除第 i 个结点，

```
struct   node
{    int   data;
      struct   node   *next;
};
typedef   struct   node   NODE
int delete(NODE *head,int i )
{
 NODE *p,*q;
 int j;
 q=head;
j=0;
 while((q!=NULL)&&( ___(1)_____))
{
___(2)_____;
j++;
}
 if(q==NULL)
 return(0);
 p= ___(3)_____;
___(4)_____=p->next;
 free(___(5)_____);
 return(1);
}
```

下面答案：（1）j<i-1
（2）q=q->next
（3）q->next
（4）q->next
（5）p

3. 栈和队列

3.1 栈的概念题

1．栈的插入删除操作在（　　）进行。　【B】
A．栈底　　B．栈顶　　C．任意位置　　D．指定位置

2．以下说法正确的是（　　）。　【C】
A．栈的特点是先进先出，队列的特点是先进后出 B．栈和队列的特点都是先进后出
C．栈的特点是先进后出，队列的特点是先进先出 D．栈和队列的特点都是先进先出

3．栈和队列的相同点是（　　）。　【D】
A．都是后进先出　　B．都是后进后出

C．逻辑结构与线性表不同

D．逻辑结构与线性表相同，都是操作规则受到限制的线性表

3.2 进栈出栈题

1．元素 3，6，9 按顺序依次进栈，则该栈的不可能输出序列是（　　）（进栈出栈可以交替进行）。

A．9，3，6　　B．9，6，3　　【A】

C．6，3，9　　D．3，9，6

2．元素 2，4，6，8 按顺序依次进栈，则该栈的不可能输出序列是（　　）（进栈出栈可以交替进行）。【D】

A．8，6，4，2　　B．2，4，6，8

C．4，2，8，6　　D．8，6，2，4

3．一个栈的进栈序列是 5，6，7，8，则栈的不可能的出栈序列是（　　）（进出栈操作可以交替进行）　【A】

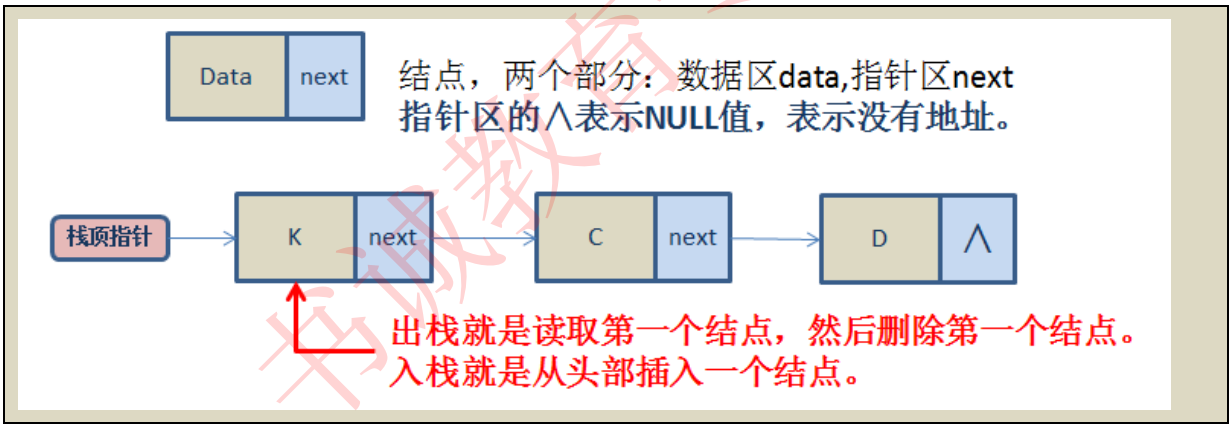
A．5，8，6，7　　B．7，6，8，5

C．7，6，5，8　　D．8，7，6，5

4．一个栈的进栈序列是 efgh，则栈的不可能的出栈序列是（　）（进出栈操作可以交替进行）。　【D】

A．hgfe　　B．gfeh　　C．fgeh　　D．ehfg

3.3 链栈指针题



1．从一个栈顶指针为 h 的链栈中删除一个结点时，用 x 保存被删结点的值，可执行 x=h->data;和_____。(结点的指针域为 next)　【h=h->next;】

2．向一个栈顶指针为 h 的链栈中插入一个 s 所指结点时，可执行_____和 h=s;。　【s->next=h;】

3．从一个栈顶指针为 h 的链栈中删除一个结点时，用 x 保存被删结点的值，可执行_____和 h=h->next;。(结点的指针域为 next)　【x=h->data;】

4．设有一个非空的链栈，栈顶指针为 hs，要进行出栈操作，用 x 保存出栈结点的值，栈结点的指针域为 next，数据域为 data，则可执行 x= _____；和 hs= _____；　【hs->data;　hs->next;】

5．设 top 是一个链栈的栈顶指针，栈中每个结点由一个数据域 data 和指针域 next 组成，设用 x 接

收栈顶元素，则出栈操作为（　　）。　【A】

A．x=top->data;top=top->next;　　B．top=top->next;x=top->data;

C．x=top-> next;top=top-> data;　　D．top->next =top; x=top->data;

6．设 top 是一个链栈的栈顶指针，栈中每个结点由一个数据域 data 和指针域 next 组成，设用 x 接收栈顶元素，则取栈顶元素的操作为（　　）。　【C】

A．top->data= x;　　B．top=top->next;

C．x=top->data;　　D．x=top->data; top= top->next;

11．C　12．C 13．D　14．B　15．A

7．设有一个链栈，栈顶指针为 hs，现有一个 s 所指向的结点要入栈，则可执行操作 s-> next=hs; _____。　【hs=s;】

8．设有一个非空的链栈，栈顶指针为 hs，要进行出栈操作，用 x 保存出栈结点的值，栈结点的指针域为 next，则可执行 x=hs->data; _____。【hs=hs->next;】

9．设有一个链栈，栈顶指针为 hs，现有一个 s 所指向的结点要入栈，则可执行操作_____ 和 hs=s;【s->next=hs;】

3.4 链栈编程题

1．**编程题** 以下函数为链栈的进栈操作，x 是要进栈的结点的数据域，top 为栈顶指针

```
struct   node
{    ElemType   data;
     struct   node   *next;
};
 struct node *top ;
  void Push(ElemType x)
 {
    struct node *p;
    p=(struct node*)malloc(___(1)_____);
    p->data=x;
   ___(2)_____;
   _____(3)___;
 }
```

下面答案：　（1）sizeof (struct node)

（2）p->next=top

（3）top=p

3.5 队列概念题

1．队列的删除操作在（　）进行。　【A】

A．队头　　B．队尾　　C．队头或队尾　　D．在任意指定位置

2．以下说法正确的是（　　）。　【C】

A．队列是后进先出　　B．栈的特点是后进后出

C．栈的删除和插入操作都只能在栈顶进行

D．队列的删除和插入操作都只能在队头进行

3．以下说法不正确的是（　　）。　【C】

A．栈的特点是后进先出　　B．队列的特点是先进先出

C．栈的删除操作在栈底进行，插入操作在栈顶进行

D．队列的插入操作在队尾进行，删除操作在队头进行

4．在队列的顺序存储结构中，当插入一个新的队列元素时，_____指针的值增 1，当删除一个元素队列时，_____指针的值增 1。【尾　头】

3.6 链队指针题

1．在一个链队中，假设 f 和 r 分别为队头和队尾指针，则删除一个结点的运算为（　　）。【D】

A．r=f->next;　B．r=r->next;　C．f=r->next;　D．f=f->next;

2．在一个链队中，设 f 和 r 分别为队头和队尾指针，则插入 s 所指结点的操作为_____和 r=s;(结点的指针域为 next)　【r->next=s;】

3．在一个链队中，f 和 r 分别为队头和队尾指针，队结点的指针域为 next，s 指向一个要入队的结点，则入队操作为_____；_____；　【r->next=s；r=s;】

4．在一个链队中，f 和 r 分别为队头和队尾指针，队结点的指针域为 next，则插入一个 s 所指结点的操作为_____；r=s；　【r->next=s】

5．栈和队列的操作特点分别是_____和 _____。

【先进后出（后进先出）　先进先出　（后进后出）】

6．在一个不带头结点的非空链队中，f 和 r 分别为队头和队尾指针，队结点的数据域为 data，指针域为 next，若要进行出队操作，并用变量 x 存放出队元素的数据值，则相关操作为_____；_____。

【x=f->data; f=f->next;】

7．**综合题**（1）设 head1 和 p1 分别是不带头结点的单向链表 A 的头指针和尾指针，head2 和 p2 分别是不带头结点的单向链表 B 的头指针和尾指针，若要把 B 链表接到 A 链表之后，得到一个以 head1 为头指针的单向循环链表，写出其中两个关键的赋值语句（不用完整程序，结点的链域为 next）。

（2）单向链表的链域为 next，设指针 p 指向单向链表中的某个结点，指针 s 指向一个要插入链表的新结点，现要把 s 所指结点插入 p 所指结点之后，某学生采用以下语句：

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

这样做正确吗？若正确则回答正确，若不正确则说明应如何改写

下面答案：

（1）p1->next= head2；p2->next= head1；

（2）不对，s->next= p->next；p->next=s；

3.7 链队编程题

1．**编程题** 以下函数为链队列的入队操作，x 为要入队的结点的数据域的值，front、rear 分别是链队列的队头、队尾指针

```
struct  node
{   ElemType  data;
    struct  node  *next;
```

```
};
struct   node   *front，*rear;

 void InQueue(ElemType x)
   {
        struct node *p;
        p= (struct node*) ___(1)_____;
        p->data=x;
        p->next=NULL;
        ___(2)_____;
        rear= ___(3)_____;
   }
```

下面答案：　（1）malloc(sizeof (struct node))

（2）rear->next=p

（3）p

2.**编程题** 以下函数为链队列的出队操作（链队列带有头结点），出队结点的数据域的值由 x 返回，front、rear 分别是链队列的队头、队尾指针

```
struct   node
{    ElemType   data;
     struct   node   *next;
};
struct   node   *front，*rear;
       ElemType OutQueue()
          {
             ElemType x;
             if(___(1)_____) {
                    printf("队列下溢错误！\n");
                    exit(1);
              }
              else {
                    struct node *p=front->next;
                    x=p->data;
                    front->next= ___(2)_____;
                    if(p->next==NULL) rear=front;
                    free(p);
                   ___(3)_____;
              }
          }
```

下面答案：　（1）front= =rear

（2）p->next

（3）return(x)

3.8 循环队列题

1．循环队列的队头指针为 f，队尾指针为 r，当____ _时表明队列为空。　【r= =f】

2．循环队列的最大存储空间为 MaxSize=6，采用少用一个元素空间以有效地判断栈空或栈满，若队头

指针 front=4，当队尾指针 rear= _____时队满，队列中共有_____个元素。 【3；5】

3．循环队列的最大存储空间为 MaxSize=8，采用少用一个元素空间以有效的判断栈空或栈满，若队头指针 front=4，则当队尾指针 rear= _____时，队列为空，当 rear= _____时，队列有 6 个元素。【4；2】

4．循环队列的引入，目的是为了克服_____ 。【假上溢】

5．循环队列的最大存储空间为 MaxSize，队头指针为 f，队尾指针为 r，当_____时表明队列已满。【（r+1）% MaxSize = f 】

4. 串

4.1 串的基本概念

性质：
“”双引号内的字符串是以‘\0’表示结束的。要多占一个字节。

1．在 C 语言中，顺序存储长度为 3 的字符串，需要占用（ ）个字节。【】
A．3 B．4 C．6 D．12

2．在 C 语言中，利用数组 a 存放字符串“Hello”，以下语句中正确的是（ ）。【A】
A．char a[10]= “Hello”； B．char a[10]; a=“Hello”；
C．char a[10]= ‘Hello’; D．char a[10]={‘H’,’e’,’l’,’l’,’o’};

3．两个串相等的充分必要条件是_____。【串长度相等且对应位置的字符相等】

4．串的两种最基本的存储方式是_____和 _____。【顺序存储 链式存储】

5．‘A‘在存储时占_____个字节。 “A”在存储时占_____个字节。 【1；2】

6. 顺序存储字符串“ABCD”需要占用_____个字节。【5】

4.2 串函数

性质
StrCmp 函数有三个结果：
第一个字符串大于第二个字符串，结果：1
第一个字符串等于第二个字符串，结果：0
第一个字符串小于第二个字符串，结果：-1
对应位置比较，根据 ASCII 码进行比较。
数字编码最小，其次大写字母，其后是小写字母。

1．串函数 StrCmp（“d”，“D”）的值为（ ）。【B】
A．0 B．1 C．-1 D．3

2．串函数 StrCmp(“abA”,”aba”)的值为（　　）。　【D】

A．1　　B．0　　C．“abAaba”　　D．-1

4.5 串的编程题

1．程序段 int count=0;　char *s=” ABCD”;

while(*s!=’\0’){s++;count++;}

执行后 count= _____　【4】

2．char *p;　【B】

p=StrCat(“ABD”,”ABC”);

Printf(“%s”,p);

的显示结果为（　　）。

A．-1　　B．ABDABC　　C．AB　　D．1

3．程序段 char *s=”aBcD”;n=0;

while(*s!=’\0’)

{　if(*s>=’a’&&*s<=’z’)n++;

s++;

}执行后 n= _____　【2】

5. 数组和广义表

5.1 数组坐标换算题

三个公式：

对称矩阵：P77

$$k=\begin{cases}\frac{i(i-1)}{2}+j & i\geq j\\ \frac{j(j-1)}{2}+i & i<j\end{cases}$$

下三角矩阵：P77

$$k=\begin{cases}\frac{i(i-1)}{2}+j+1 & i\geq j\\ 0 & i<j\end{cases}$$

对角矩阵：P78

$$k=\begin{cases}2i+j+1 & (i=j+1,i=j,i+1=j)\\ 0 & \text{其他}\end{cases}$$

1．设有一个 10 阶的对称矩阵 A，采用压缩存储的方式，将其下三角部分以行序为主存储到一维数组 B 中（数组下标从 1 开始），则矩阵中元素 A8,5 在一维数组 B 中的下标是（　　）。【A】

A．33　　B．32　　C．85　　D．41

2．设有一个 15 阶的对称矩阵 A，采用压缩存储的方式，将其下三角部分以行序为主序存储到一维数组 B 中（数组下标从 1 开始），则矩阵中元素 a7,6 在一维数组 B 中的下标是（　　）。【C】

A．42　　B．13　　C．27　　D．32

3．设有一个 15 阶的对称矩阵 A，采用压缩存储方式将其下三角部分以行序为主序存储到一维数组 b 中。(矩阵 A 的第一个元素为 a1,1，数组 b 的下标从 1 开始)，则数组元素 b[13]对应 A 的矩阵元素是（　　）。　【A】

A．a5,3　　B．a6,4　　C．a7,2　　D．a6,8

4．设有 n 阶对称矩阵 A，用数组 S 进行压缩存储，当 i<j 时，A 的数组元素 aij 相应于数组 S 的数组元素的下标为_____。(数组元素的下标从 1 开始)　【i(i-1)/2+j 】

5．设有一个 12 阶的对称矩阵 A，采用压缩存储方式将其下三角部分以行序为主序存储到一维数组 b 中（矩阵 A 的第一个元素为 a1,1，数组 b 的下标从 1 开始)，则矩阵 A 中第 4 行的元素在数组 b 中的下标 i 一定有（　　）。　【A】

A、7≤i≤10　　B、11≤i≤15　　C、9≤i≤14　　D、6≤i≤9

5.2 矩阵题

1．稀疏矩阵存储时，采用一个由____ 、____ 、____3 部分信息组成的三元组唯一确定矩阵中的一个非零元素。【行号；列号；非零元】

6. 树和二叉树

6.1 二叉树的概念性质

P92 二叉树的性质

1. 二叉树上终端结点数等于双分支结点数加 1。

2. 二叉树上第 i 层上至多有 2^{i-1} 个结点（$i \geq 1$)。

3. 深度为 h 的二叉树至多有 2^h-1 个结点。

4. 对于一颗二叉树中顺序编号为 i 的结点，若它存在左孩子，则左孩子结点的编号为2i；若它存在

孩子，则右孩子结点的编号为 2i+1，若它存在双亲结点（即编号不等于 1)，则双亲结点的编号为$\lfloor i/2$

5. 具有 n 个结点的理想二叉树的深度为$\lceil log_2(n+1) \rceil$ 或 $\lfloor log_2 n \rfloor + 1$　。

1．在一棵二叉树中，若编号为 i 的结点存在右孩子，则右孩子的顺序编号为（　　）。　【D】

A．2i　　B．2i-1　　C．2i+2　　D．2i+1

2．一棵二叉树中顺序编号为 i 的结点，若它存在左、右孩子，则左、右孩子编号分别为_____、_____。

【2i 和 2i+1】

注释：1 题和 2 题都是利用性质 4

3．一棵有 n 个叶结点的二叉树，其每一个非叶结点的度数都为 2，则该树共有_____个结点。

【2n-1】

4．一棵有 2n-1 个结点的二叉树，其每一个非叶结点的度数都为 2，则该树共有_____个叶结点。

【n 】

注释：３题和４题是利用同一个性质 1 完成。

结点总数：$n = n_0 + n_1 + n_2$

每一个非叶结点的度数都为 2，说明：$n_1 = 0$，就是结点要么没有孩子，要么有两个孩子。

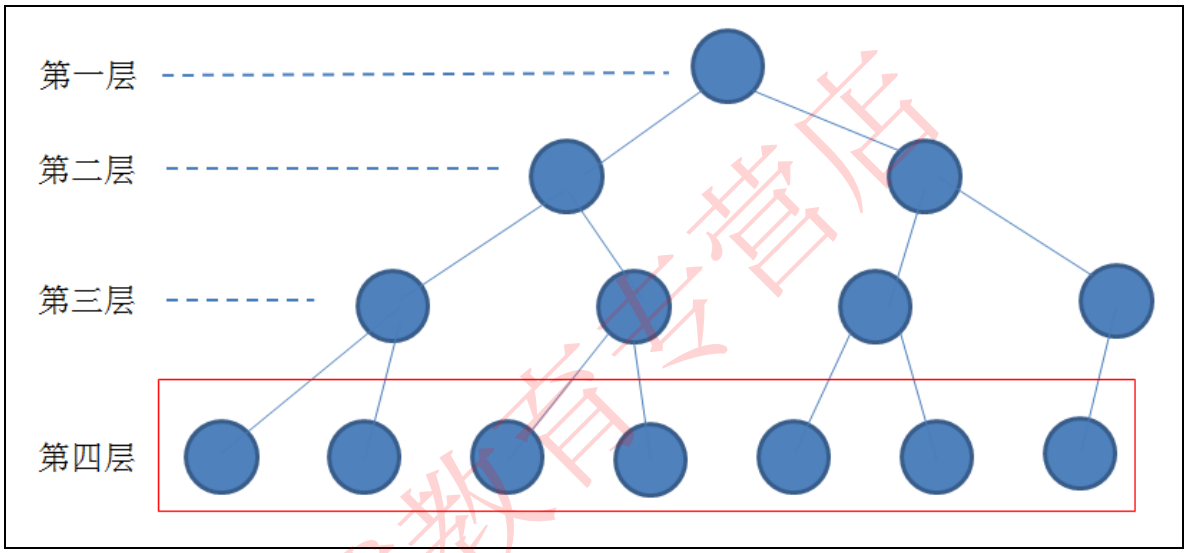
结果：总结点数$n = n_0 + n_2$

根据性质 1 又有：$n_0 = n_2 + 1$

根据上面两个结论，可以求出 3 题和 4 题。

5．一棵有 14 个结点的完全二叉树，则它的最高层上有_____个结点。 【7】

注释：完全二叉树如下图所示：最高一层数一下有七个结点。



6. 综合题之一 “一棵二叉树若它的根结点的值大于左子树所有结点的值，小于右子树所有结点的值，则该树一定是二叉排序树”。该说法是否正确，若认为正确，则回答正确，若认为不正确则说明理由？ 【不正确，二叉排序树要求其子树也是二叉排序树。】

7．一棵完全二叉树共有 30 个结点，则该树一共有（ ）层(根结点所在层为第一层)。【D】

A．6 B．4 C．3 D．5

注释：两个方法：1. 画图 2. 套公式：性质 5 的公式。

8．一棵二叉树总结点数为 11，叶结点数为 5，该树有_____个双分支结点，_____个单分支结点。

【4；2】

注释：和３题４题一样，是利用同一个性质 1 完成。参阅前面的注释。

9．深度为 k 的二叉树最多有_____结点。 【2^k-1 】

10．深度为 5 的满二叉树至多有（ ）个结点（根结点为第一层） 【B】

A．40 B．31 C．34 D．35

注释：9 题和 10 题是套性质 3 的公式 $2^h - 1$

11．一棵二叉树中顺序编号为 5 的结点（树中各结点的编号与等深度的完全二叉中对应位置上结点的编号相同），若它存在左孩子，则左孩子的编号为_____。 【10】

12. 一棵二叉树没有单分支结点，有 6 个叶结点，则该树总共有_____个结点。【11】

13. 一棵有 n 个叶结点的二叉树，其每一个非叶结点的度数都为 2，则该树共有_____个结点。【2n-1】

14. 一棵二叉树叶结点（终端结点）数为 5，单分支结点数为 2，该树共有_____个结点。【11】

注释：12 题 13 题 14 题　和３题４题一样，是利用同一个性质 1 完成。参阅前面的注释。

15. 二叉树为二叉排序的充分必要条件是其任一结点的值均大于其左孩子的值、小于其右孩子的值。这种说法是_____的。(回答正确或不正确)　【错误】

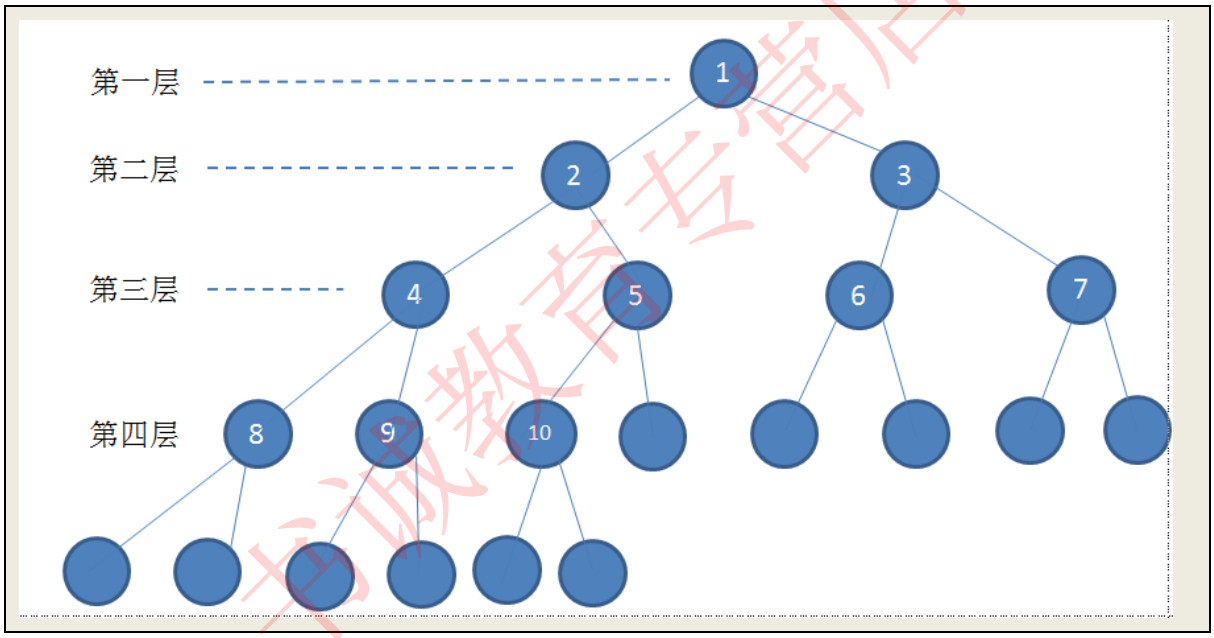
注释：论断太绝对，叶子结点没有后代。

16. 一棵二叉树顺序编号为 6 的结点（树中各结点的编号与等深度的完全二叉中对应位置上结点的编号相同），若它存在右孩子，则右孩子的编号为_____。【13】

注释：利用性质 4

17. 设一棵完全二叉树，其最高层上最右边的叶结点的编号为奇数，该叶节点的双亲结点的编号为 10，该完全二叉树一共有_____个结点。【21】

注释：参阅下图：



6.2 二叉树的链式存储

1. 设一棵有 n 个结点采用链式存储的二叉树，则该树共有（　　）个指针域为空。　【B】

A. 2n　　B. n+1　　C. 2n+1　　D. 2n+2

6.3 树的遍历概念题

1. 对二叉排序树进行（　　）遍历，遍历所得到的序列是有序序列。　【C】

A. 按层次　　B. 前序　　C. 中序　　D. 后序

2. _____遍历二叉排序树可得到一个有序序列。　【中序 】

3. 对二叉树的遍历可分为_____、_____、_____、_____四种不同的遍历次序。

【先序、中序、后序、层次】

4．设一棵完全二叉树，其最高层上最右边的叶结点的编号为偶数，该叶节点的双亲结点的编号为 9，该完全二叉树一共有_____个结点。 【18】

5．一棵有 n 个叶结点的二叉树，其每一个非叶结点的度数都为 2，则该树共有_____个结点。【2n-1】

6．一棵哈夫曼树总共有 23 个结点，该树共有（ ）个叶结点（终端结点） 【D】

A．10 B．13 C．11 D．12

注释：本题和３题４题一样，是利用同一个性质１完成。参阅前面的注释。

本题由“哈夫曼树”隐含的说明了一个条件：没有单分支的结点。

7．一棵哈夫曼树总共有 25 个结点，该树共有（ ）个非叶结点（非终端结点）。【A】

A．12 B．13 C．14 D．15

8．按照二叉树的递归定义，对二叉树遍历的常用算法有_____ 、_____、 _____三种。【先序；中序；后序】

6.4 树的遍历操作题

1．如图 2 所示的二叉树，其先序遍历序列为_____。 【abdgcefhi】

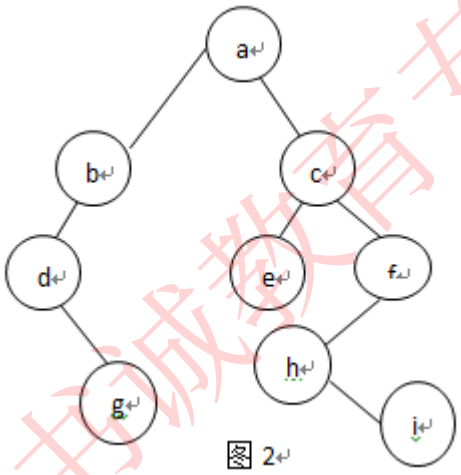
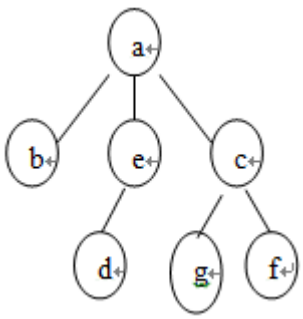


图 2

2．如图若从顶点 a 出发按深度优先搜索法进行遍历，则可能得到的顶点序列为（ ）。【A】

A．acfgedb B．aedcbgf C．acfebdg D．aecbdgf



3．如图若从顶点 a 出发按广度优先搜索法进行遍历，则可能得到的顶点序列为（ ）。 【D】

A．acebdfgh B．aebcghdf C．aedfbcgh D．abecdfgh

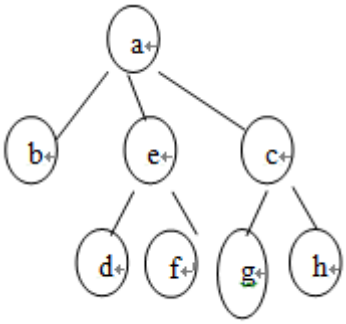


图 1

4．如图 2 所示的二叉树，其后序遍历序列为_____。【gdbeihfca】

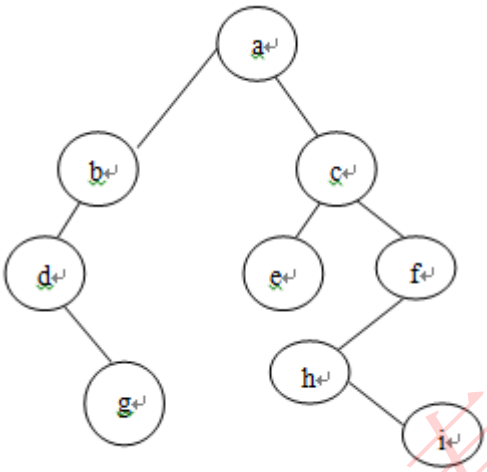


图 2

5．如图 2 所示的二叉树，其前序遍历序列为_____。【abdefcg】

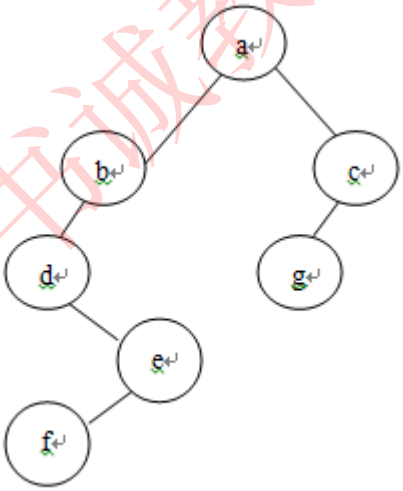


图 2

6．如图 2 所示的二叉树，其后序遍历序列为_____。【gdbeihfca】

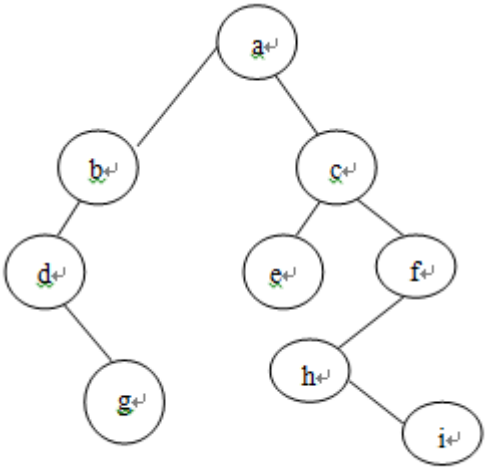
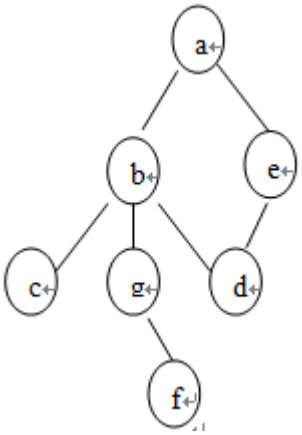


图 2

7．如图若从顶点 a 出发按深度优先搜索法进行遍历，则可能得到的顶点序列为（　　）。　【B】

A．acfgedb　　B．aedbgfc　　C．acfebdg　　D．aecbdgf



6.5 树的遍历编程题

1．**编程题** 以下程序是先序遍历二叉树的递归算法的程序，完成程序中空格部分（树结构中左、右指针域分别为 left 和 right，数据域 data 为字符型，BT 指向根结点）。

```
void Preorder (struct BTreeNode *BT)
{    if(BT!=NULL){
      （1）_____;
      （2）_____;
      （3）_____;
      }
}
```

下面答案：　（1）printf(“%c”,BT->data)

（2）Preorder(BT->left)

（3）Preorder(BT->right)

2．**编程题** 以下程序是后序遍历二叉树的递归算法的程序，完成程序中空格部分（树结构中，左、右指针域分别为 left 和 right，数据域 data 为字符型，BT 指向根结点）。

```
void Postorder (struct BTreeNode *BT)
{    if(BT!=NULL){
      （1）_____;
```

（2）_____;
（3）_____;
}
}

下面答案： （1）Postorder(BT->left)
（2）Postorder(BT->right)
（3） printf(“%c”,BT->data)

3．**编程题** 以下程序是中序遍历二叉树的递归算法的程序，完成程序中空格部分（树结构中左、右指针域分别为 left 和 right，数据域 data 为字符型，BT 指向根结点）。

```
void Inorder (struct BTreeNode *BT)
{   if(BT!=NULL){
      （1）_____;
      （2）_____;
      （3）_____;
      }
}
```

下面答案： （1） Inorder(BT->left)
（2）printf(“%c”,BT->data)
（3） Inorder(BT->right)

6.6 哈夫曼树

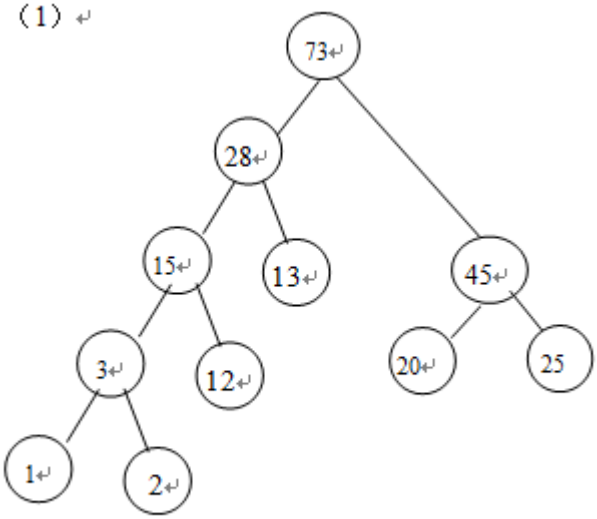
1．设一棵哈夫曼树共有 n 个叶结点，则该树有（ ）个非叶结点。 【A】
A．n-1 B．n C．n+1 D．2n

2．一棵哈夫曼树有 12 个叶子结点（终端结点），该树总共有（ ）个结点。 【C】
A．22 B．21 C．23 D．24

3．（1）以给定权重值 1，2，12，13，20，25 为叶结点，建立一棵哈夫曼树
（2）若哈夫曼树有 n 个非叶子结点，则树中共有多少结点。对给定的一组权重值建立的棵哈夫曼树是否一定唯一

下面答案

（1）



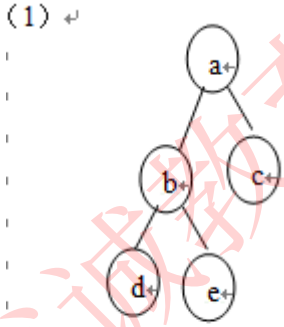
（2）2n-1　不一定唯一

6.7 树的遍历反过来做的题

1. **综合题** （1）已知某二叉树的后序遍历序列是 debca，中序遍历序列是 dbeac，试画出该二叉树
（2）若上述二叉树的各个结点的字符分别代表不同的整数（其中没有相等的），并恰好使该树成为一棵二叉排序树，试给出 a、b、c、d、e 的大小关系
（3）给出该树的前序遍历序列

下面答案：

（1）



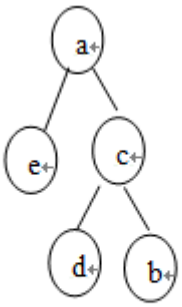
（2）d<b<e<a<c

（3）abdec

2. **综合题** （1）已知某二叉树的先序遍历序列是 aecdb，中序遍历序列是 eadcb，试画出该二叉树
（2）给出上述二叉树的后序遍历序列
（3）若上述二叉树的各个结点的字符分别是 1，2，3，4，5，并恰好使该树成为一棵二叉排序树，试问 a、b、c、d、e 的值各为多少？

下面答案：

（1）



（2）edbca

（3）e=1,a=2,d=3,c=4,b=5

7. 图

7.1 图的基本概念

P124:
全部顶点的度数为所有边数的 2 倍，或者说，边数为全部顶点的度数的一半。

1．在一个无向图中，所有顶点的度数之和等于边数的（　　）倍。【D】
A．3　　B．2.5　　C．1.5　　D．2

2．已知一个图的边数为 m，则该图的所有顶点的度数之和为（　　）。【A】
A．2m　　B．m　　C．2m+1　　D．m/2

3．已知一个图的所有顶点的度数之和为 m，则该图的边数为（　　）。【D】
A．2m　　B．m　　C．2m+1　　D．m/2

4．以下说法不正确的是（　　）。【D】
A．连通图 G 一定存在生成树
B．连通图 G 的生成树中一定包含 G 的所有顶点
C．连通图 G 的生成树中不一定包含 G 的所有边
D．连通图 G 的生成树可以是不连通的

5．以下说法不正确的是（　　）。【A】
A．连通图 G 的生成树一定是唯一的
B．连通图 G 一定存在生成树
C．连通图 G 的生成树中一定要包含 G 的所有顶点
D．连通图 G 的生成树一定是连通而且不包含回路

7.2 图的遍历

1．根据搜索方法的不同，图的遍历有____、 ____ 两种方法 【深度优先　广度优先】

2．图的深度优先搜索和广度优先搜索序列不一定是唯一的。此断言是_____的。(回答正确或不正确)
【正确】

3．已知如图 1 所示的一个图，若从顶点 a 出发，按广度优先搜索法进行遍历，则可能得到的一种顶点序列为（　　）。 【C】

A．abcedf　　B．aebcfd　　C．abcefd　　D．acfdeb

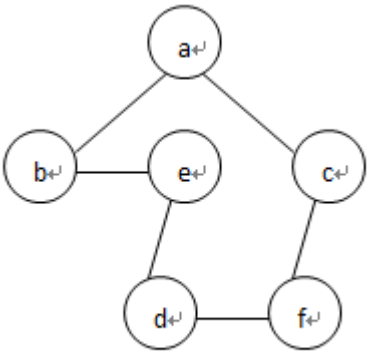


图 1

4．已知如图 1 所示的一个图，若从顶点 V_1 出发，按广度优先进行遍历，则可能得到的一种顶点序列为（　　）。 【C】

A．$V_1V_2V_3V_6V_7V_4V_5V_8$　　B．$V_1V_2V_3V_4V_5V_8V_6V_7$

C．$V_1V_2V_3V_4V_5V_6V_7V_8$　　D．$V_1V_2V_3V_4V_8V_5V_6V_7$

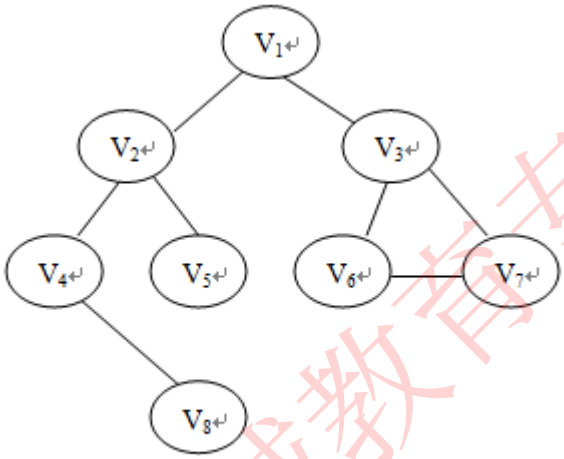
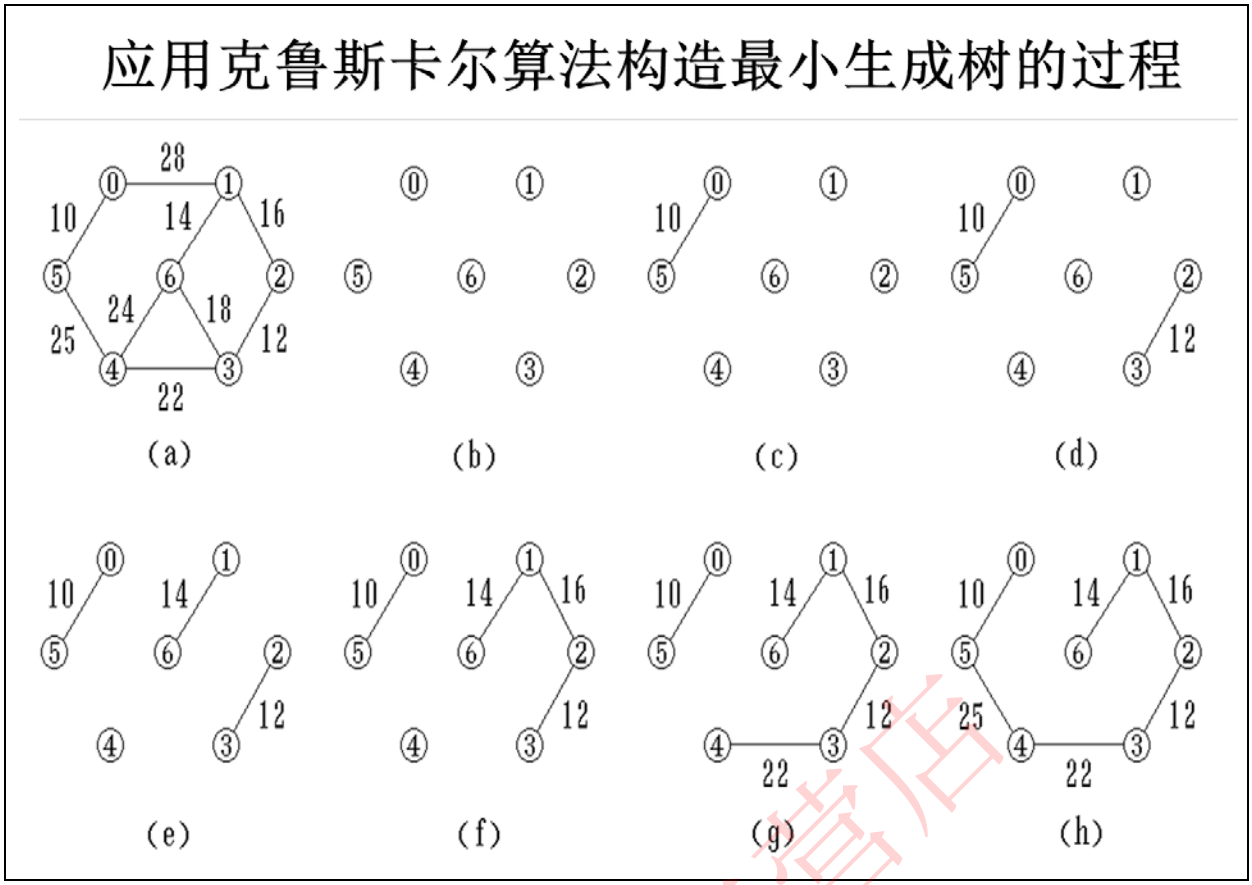


图 1

7.3 图的最小生成树

应用克鲁斯卡尔算法构造最小生成树的过程



7.5 图的连通性

1．以下说法正确的是（　　）。　【D】

A．连通图 G 的生成树中不一定包含 G 的所有顶点

B．连通图 G 的生成树中一定要包含 G 的所有边

C．连通图 G 的生成树一定是唯一的

D．连通图 G 一定存在生成树

8. 查找

8.1 顺序查找

1．对长度为 n 的线性表进行顺序查找，在等概率情况下，平均查找长度为（　　）。　【B】

A．n　　B．（n+1）/2　　C．2n　　D．n-1

2．采用顺序查找法对长度为 n 的线性表进行查找（不采用表尾设监视哨的方法），最坏的情况下要进行（　　）次元素间的比较。　【B】

A．n+2　　B．n　　C．n-1　　D．n/2

8.2 折半查找

1．在有序表{2，4，7，14，34，43，47，64，75，80，90，97，120}中，用折半查找法查找值 80 时，经（　　）次比较后查找成功。　【】

A．2　　B．3　　C．4　　D．5

2．有一个长度为 10 的有序表，按折半查找对该表进行查找，在等概率情况下查找成功的平均比较次数为（ A ）。

A．29/10　　B．31/10　　C．26/10　　D．29/9

3．在有序表{1，3，8，13，33，42，46，63，76，78，86，97，100}中，用折半查找值 86 时，经（ ）次比较后查找成功。 【D】

A．6　　B．3　　C．8　　D．4

4．用折半查找法，对长度为 12 的有序的线性表进行查找，最坏情况下要进行（ ）次元素间的比较 【A】

A．4　　B．3　　C．5　　D．6

5．**综合题** 设有序表为（13，19，25，36，48，51，63，84，91，116，135，200），元素的下标依次为 1,2,……,12。 （1）说出有哪几个元素需要经过 4 次元素间的比较才能成功查到

（2）画出对上述有序表进行折半查找所对应的判定树（树结点用下标表示）

（3）设查找元素 5，需要进行多少次元素间的比较才能确定不能查到

下面答案：

(1)19，48，84，116，200

(2)

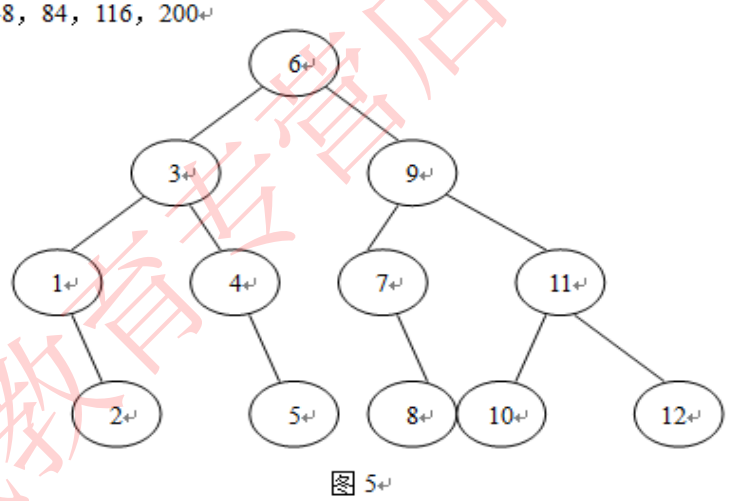


图 5

(3)3 次

6．折半查找只适用于_____存储的有序表 。【顺序存储结构】

7．有序表为{1，2，4，6，10，18，20，32}，用课本中折半查找算法查找值 18，经（ ）次比较后成功查到。 【B】

A．3　　B．2　　C．4　　D．5

8.3 二叉排序树

1．二叉树为二叉排序的充分必要条件是其任一结点的值均大于其左孩子的值、小于其右孩子的值。这种说法是_____的。(回答正确或不正确) 【不正确】

2．**综合题** （1）对给定数列{7,16,4,8,20,9,6,18,5}，依次取数列中的数据，构造一棵二叉排序树。

(2)对一个给定的查找值,简述针对二叉排序树进行查找的算法步骤，在上述二叉树中查找元素 20 共要进行多少次元素的比较?

下面答案：

(1)

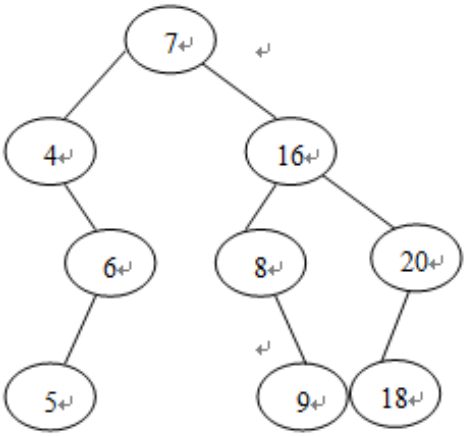


图 6

(2)先将给定值与根结点比较，若相等则查找成功，否则若小于根结点则在左子树中继续查找，大于根结点在右子树中查找，查找 20 共进行 3 次比较。

3．**综合题**

(1)“一棵二叉树若它的根结点的值大于左子树所有结点的值，小于右子树所有结点的值，则该树一定是二叉排序树”。该说法是否正确，若认为正确，则回答正确，若认为不正确则说明理由？

(2)设有查找表{7，16，4，8，20，9，6，18，5}，依次取表中数据构造一棵二叉排序树. 对上述二叉树给出后序遍历的结果.

下面答案：

(1)不正确，二叉排序树要求其子树也是二叉排序树。

(2)

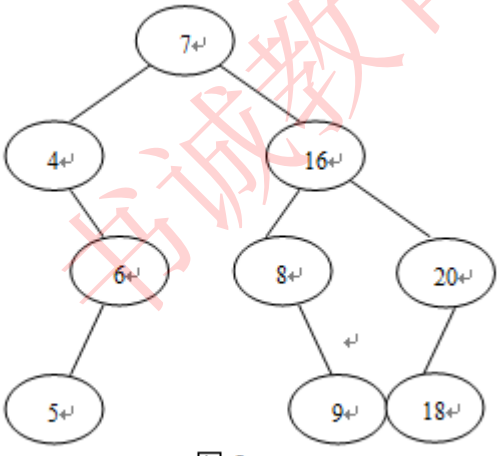


图 5

后续遍历 5，6，4，9，8，18，20，16，7

4．二叉树排序中任一棵子树都是二叉排序树，这种说法是_____的。(回答正确或不正确)

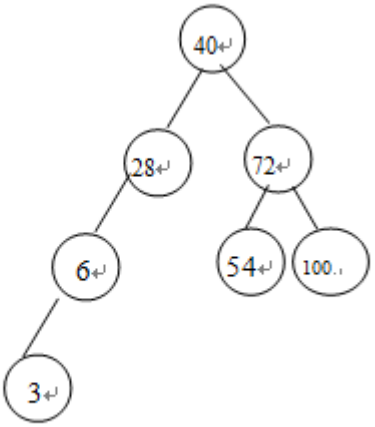
【正确】

5．**综合题**（1）设有一个整数序列{40，28，6，72，100，3，54}依次取出序列中的数，构造一棵二叉排序树

（2）对上述二叉排序树，在等概率条件下，求成功查找的平均查找长度

下面答案：

（1）



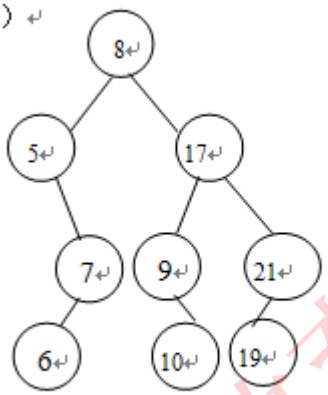
（2）ASL=（1x1+2x2+3x3+4）/7=18/7

6．**综合题**（1）给定数列{8，17，5，9，21，10，7，19，6}，依次取序列中的数构造一棵二叉排序树

（2）对上述二叉树给出中序遍历得到的序列。

下面答案：

（1）



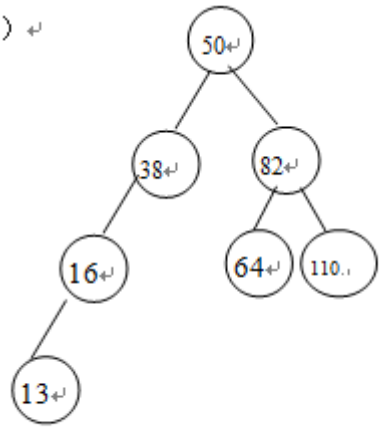
（2）5，6，7，8，9，10，17，18，19，21

7．**综合题**（1）设有一个整数序列{50，38，16，82，110，13，64}，依次取出序列中的数，构造一棵二叉排序树

（2）利用上述二叉排序树，为了查找 110，经多少次元素间的比较能成功查到，为了查找 15，经多少次元素间的比较可知道查找失败

下面答案：

（1）



（2）三次；四次

8.4 二叉判定树

解析：写判定树最重要是每次找子树对应的顶点

1．综合题 设查找表为(16,15,20,53,64,7),

(1)用冒泡法对该表进行排序（要求升序排列），写出每一趟的排序过程，通常对 n 个元素进行冒泡排序要进行多少趟冒泡？第 j 趟要进行多少次元素间的比较？

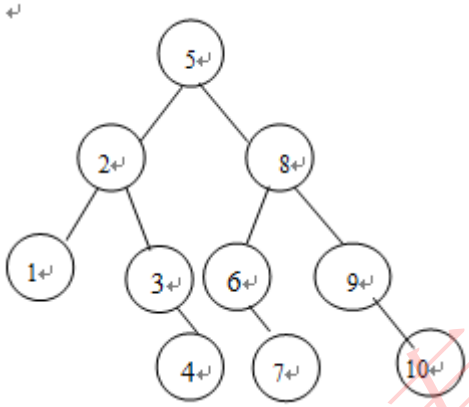
(2)在排序后的有序表的基础上，画出对其进行折半查找所对应的判定树.(要求以数据元素作为树结点)

(3)求在等概率条件下,对上述有序表成功查找的平均查找长度.

2．综合题（1）画出对长度为 10 的有序表进行折半查找的判定树（以序号 1，2，……10 表示树结点）（2）对上述序列进行折半查找，求等概率条件下，成功查找的平均查找长度

下面答案：

（1）



（2）ASL=（1x1+2x2+3x4+4x3）/10=29/10

8.5 哈希函数

1．哈希函数是记录关键字值与该记录存储地址之间所构造的_____。 【对应关系】

2．哈希函数是记录关键字值与该记录_____之间所构造的对应关系。 【存储地址】

3．散列查找的原理是（ ）。 【A】

A．在待查记录的关键字值与该记录的存储位置之间建立确定的对应关系

B．按待查记录的关键字有序的顺序方式存储

C．按关键字值的比较进行查找

D．基于二分查找的方法

8.6 折半查找编程题

1．综合题 以下函数在a[0]到a[n-1]中，用折半查找算法查找关键字等于k的记录，查找成功返回该记录的下标，失败时返回-1，完成程序中的空格

```
typedef    struct
{    int   key;
     ……
}NODE;
int Binary_Search(NODE a[],int n, int k)
  {
```

```
int low,mid,high;
low=0;
high=n-1;
while(___(1)_____)
{
    mid=(low+high)/2;
    if(a[mid].key==k)
        return __(2)_____;
    else if(___(3)_____)
        low=mid+1;
    else __(4)_____;
  }
  ___(5)_____;
}
```

下面答案：
（1）low<=high
（2）mid
（3）a[mid].key<k;
（4）high=mid-1
（5）return -1;

8.7 二叉排序树编程题

1．**编程题** 以下函数是二叉排序树的查找算法，若二叉树为空，则返回根结点的指针，否则，返回值是指向树结点的结构指针 p（查找成功 p 指向查到的树结点，不成功 p 指向为 NULL）完成程序中的空格

```
typedef  struct  Bnode
{   int  key;
    struct   Bnode *left;
    struct   Bnode *right;
} Bnode;
Bnode *BSearch(Bnode  *bt, int k)
/* bt用于接收二叉排序树的根结点的指针，k用以接收要查找的关键字*/
 {  Bnode *p;
    if(bt== ___(1)_____)
     return (bt);
    p=bt;
    while(p->key!= __(2)_____)
     {  if(k<p->key)
          ___(3)_____;
        else ___(4)_____;
        if(p==NULL) break;
     }
    Return(___(5)_____);
  }
```

下面答案：
（1）NULL
（2）k

（3）p=p->left
（4）p=p->right
（5）p

9. 排序

9.1 排序基本概念

1．按某关键字对记录序列排序，若_____在排序前和排序后仍保持它们的前后关系，则排序算法是稳定的，否则是不稳定的。 【 关键字相等的记录 】

2．以下排序算法中，在一趟排序过程中，除了其它相关操作外，只进行一次元素间的交换的算法是（ ）。【A】

A．直接选择 B．冒泡 C．直接插入 D．折半插入

9.2 直接插入排序

1．排序算法中，从未排序序列中依次取出元素与已排序序列（初始为空）中的元素进行比较（要求比较次数尽量少），然后将其放入已排序序列的正确位置的方法是（ ）。【D】

A．冒泡 B．直接插入 C．选择排序 D．折半插入

3．排序方法中，从尚未排序序列中挑选元素，并将其依次放入已排序序列（初始为空）的一端的方法，称为（ ）排序。【C】

A．归并 B．插入 C．选择 D．快速

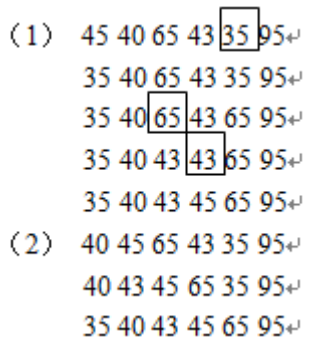
4．排序过程中，每一趟从无序子表中将一个待排序的记录按其关键字的大小放置到已经排好序的子序列的适当位置，直到全部排好序为止，该排序算法是( )。 【A】

A．直接插入排序 B．快速排序
C．冒泡排序 D．选择排序

5．综合题（1）一组记录的关键字序列为{45，40，65，43，35，95}写出利用快速排序的方法，以第一个记录为基准得到的一趟划分的结果（要求给出一趟划分中每次扫描和交换的结果）

（2）同样对序列{45，40，65，43，35，95}利用直接插入排序，写出逐次插入过程（从第一个元素一直到第六个元素）。

下面答案：



9.3 折半插入排序

9.4 交换排序之冒泡排序

1．在排序过程中，可以通过某一趟排序的相关操作所提供的信息，判断序列是否已经排好序，从而可以提前结束排序过程的排序算法是(　　)。　【A】

A．冒泡　　B．选择　　C．直接插入　　D．折半插入

2．**综合题** 设查找表为(16,15,20,53,64,7),

(1)用冒泡法对该表进行排序（要求升序排列），写出每一趟的排序过程，通常对 n 个元素进行冒泡排序要进行多少趟冒泡？第 j 趟要进行多少次元素间的比较？

(2)在排序后的有序表的基础上，画出对其进行折半查找所对应的判定树.(要求以数据元素作为树结点)

(3)求在等概率条件下,对上述有序表成功查找的平均查找长度.

下面答案：

(1)原序列 16 15 20 53 64 7

15 16 20 53 7 64　　n-1 趟

15 16 20 7 53 64　　n-j 次

15 16 7 20 53 64

15 7 16 20 53 64

7 15 16 20 53 64

(2)

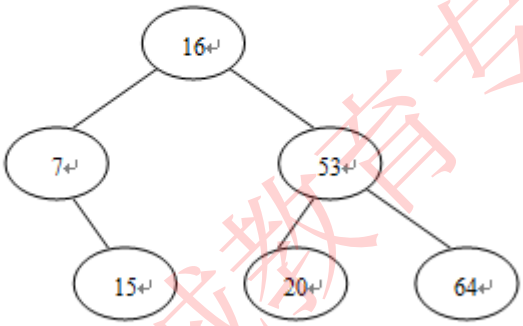


图 4

(3)平均查找长度=（1*1+2*2+3*3）/6=14/6

3．**编程题** 以下冒泡法程序对存放在 a[1]，a[2]，……，a[n]中的序列进行排序，完成程序中的空格部分，其中 n 是元素个数，要求按升序排列。

```
void bsort (NODE  a[ ], int n)
 { NODE temp;
  int i,j,flag;
  for(j=1;  (1)              ;j++);
   {flag=0;
     for(i=1;  (2)            ;i++)
       if(a[i].key>a[i+1].key)
       {flag=1;
        temp=a[i];
         (3)               ;
         (4)               ;
       }
```

```
    if(flag= =0)break;
    }
}
```

程序中 flag 的功能是（5）_____

下面答案：

（1）j<=n-1

（2）i<=n-j

（3）a[i]=a[i+1]

（4）a[i+1]=temp

（5）当某趟冒泡中没有出现交换则已排好序，结束循环

9.5 交换排序之快速排序

1．**综合题** 一组记录的关键字序列为（46，79，56，38，40，84）

（1）利用快速排序的方法，给出以第一个记录为基准得到的一次划分结果（给出逐次交换元素的过程，要求以升序排列）

（2）对上述序列用堆排序的方法建立大根堆，要求以二叉树逐次描述建堆过程。

下面答案：

(1)初始序列

46，79，56，38，40，84

40，79，56，38，[40]，84

40，[79]，56，38，79，84

40，38，56，[38]，79，84

40，38，[56]，56，79，84

40，38，46，56，79，84

(2)

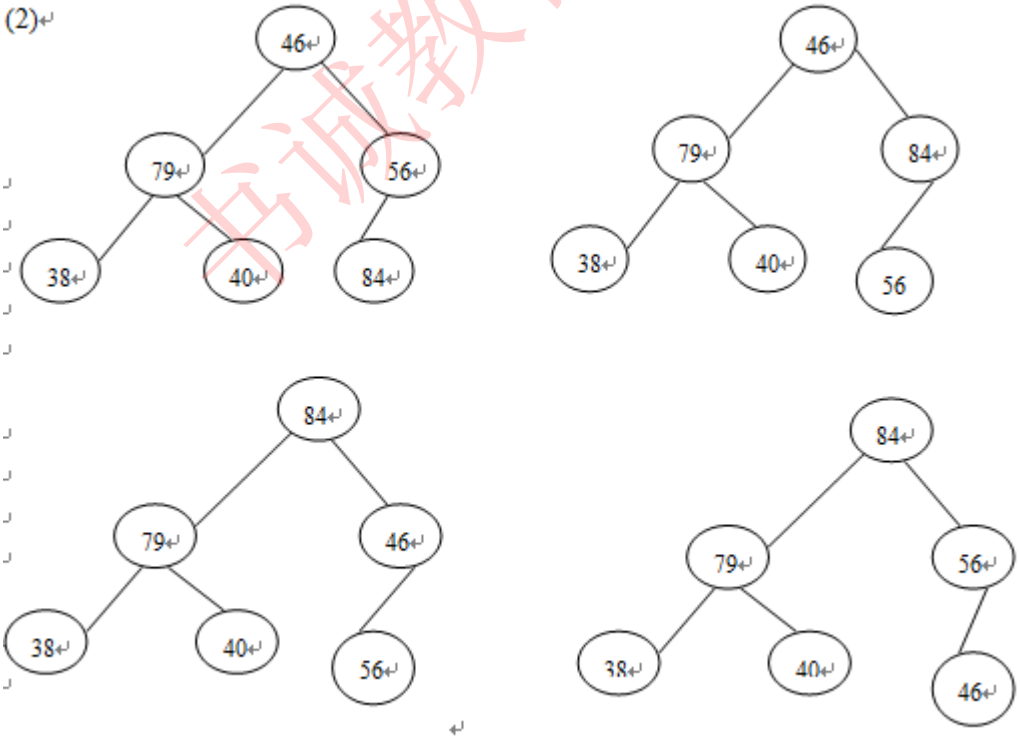


图 3

2．一组记录的关键字序列为（37，70，47，29，31，85），利用快速排序，以第一个关键字为分割元素，经过一次划分后结果为（　　）。 【A】

A．31，29，37，47，70，85　　B．29，31，37，47，70，85

C．31，29，37，70，47，85　　D．31，29，37，85，47，70

3．**综合题**（1）一组记录的关键字序列为{45，40，65，43，35，95}写出利用快速排序的方法，以第一个记录为基准得到的一趟划分的结果（要求给出一趟划分中每次扫描和交换的结果）

（2）同样对序列{45，40，65，43，35，95}利用直接插入排序，写出逐次插入过程（从第一个元素一直到第六个元素）。

下面答案：

（1） 45 40 65 43 35 95↵
35 40 65 43 35 95↵
35 40 65 43 65 95↵
35 40 43 43 65 95↵
35 40 43 45 65 95↵

（2） 40 45 65 43 35 95↵
40 43 45 65 35 95↵
35 40 43 45 65 95↵

9.6 选择排序之直接选择排序

1．**编程题** 以下函数为直接选择排序算法，对 a[1],a[2],…a[n]中的记录进行直接选择排序，完成程序中的空格

```
typedef     struct
{     int    key;
      ……
}NODE;
 void selsort(NODE a[],int n)
{
 int i,j,k;
 NODE temp;
 for(i=1;i<= ___(1)_____;i++)
 {
       k=i;
       for(j=i+1;j<= ___(2)_____;j++)
            if(a[j].key<a[k].key) __(3)_____;
       if(i!=k)
       {
            temp=a[i];
            ___(4)_____;
            _____(5)____;
       }
 }
}
```

下面答案： （1）n-1

（2）n

（3）k=j

（4）a[i]=a[k]

（5）a[k]=temp

9.7 选择排序之堆排序

1．**综合题** 设一组记录的关键字序列为（49，83，59，41，43，47），采用堆排序算法完成以下操作：（要求小根堆，并画出中间过程）

（1）以二叉树描述 6 个元素的初始堆

（2）以二叉树描述逐次取走堆顶元素后，经调整得到的 5 个元素、4 个元素的堆

下面答案：

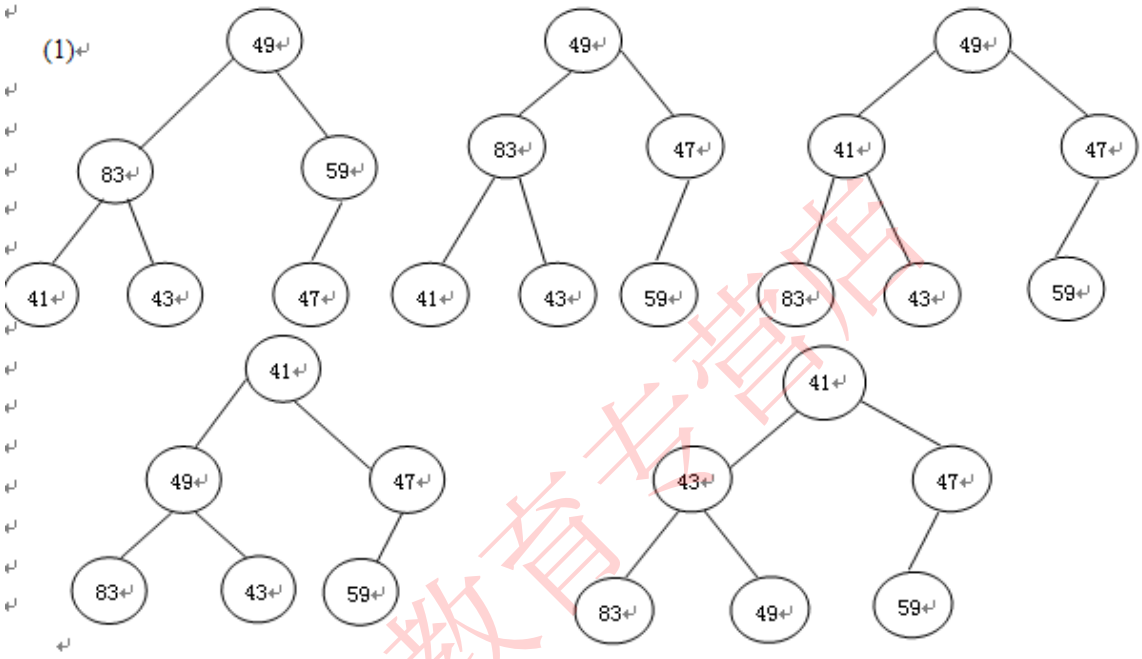


图 3

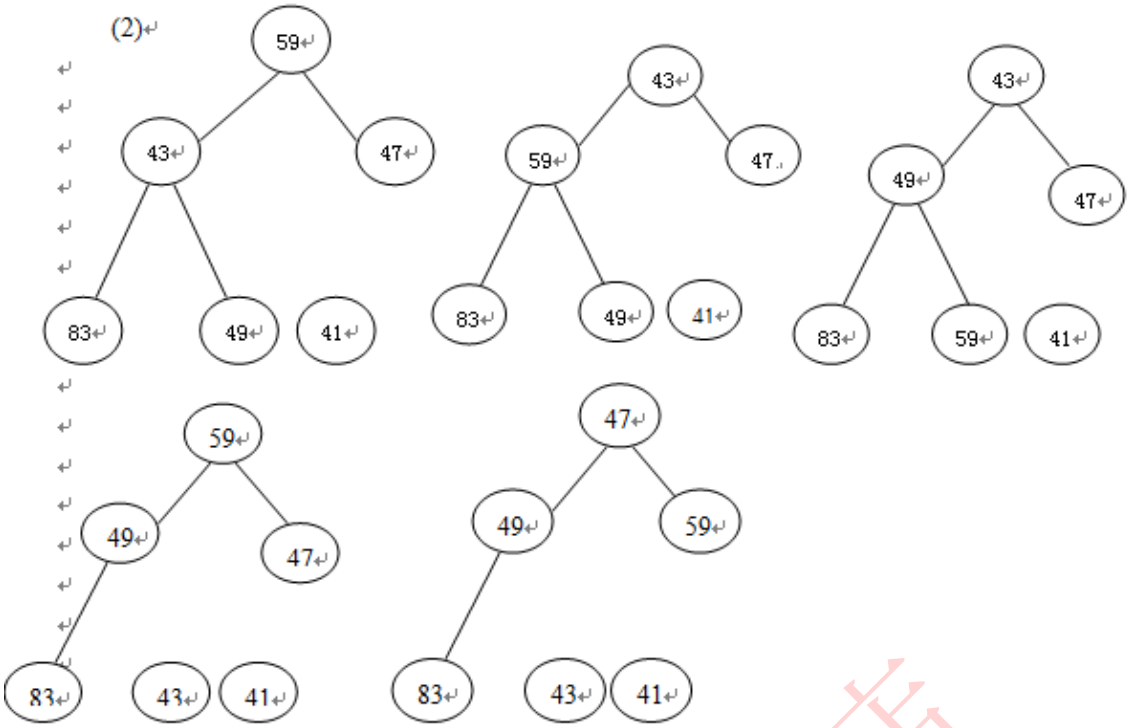


图 4

2．**综合题** 一组记录的关键字序列为（46，79，56，38，40，84）

（1）利用快速排序的方法，给出以第一个记录为基准得到的一次划分结果（给出逐次交换元素的过程，要求以升序排列）

（2）对上述序列用堆排序的方法建立大根堆，要求以二叉树逐次描述建堆过程。

下面答案：

(1)初始序列

46，79，56，38，40，84

40，79，56，38，[40]，84

40，[79]，56，38，79，84

40，38，56，[38]，79，84

40，38，[56]，56，79，84

40，38，46，56，79，84

(2)

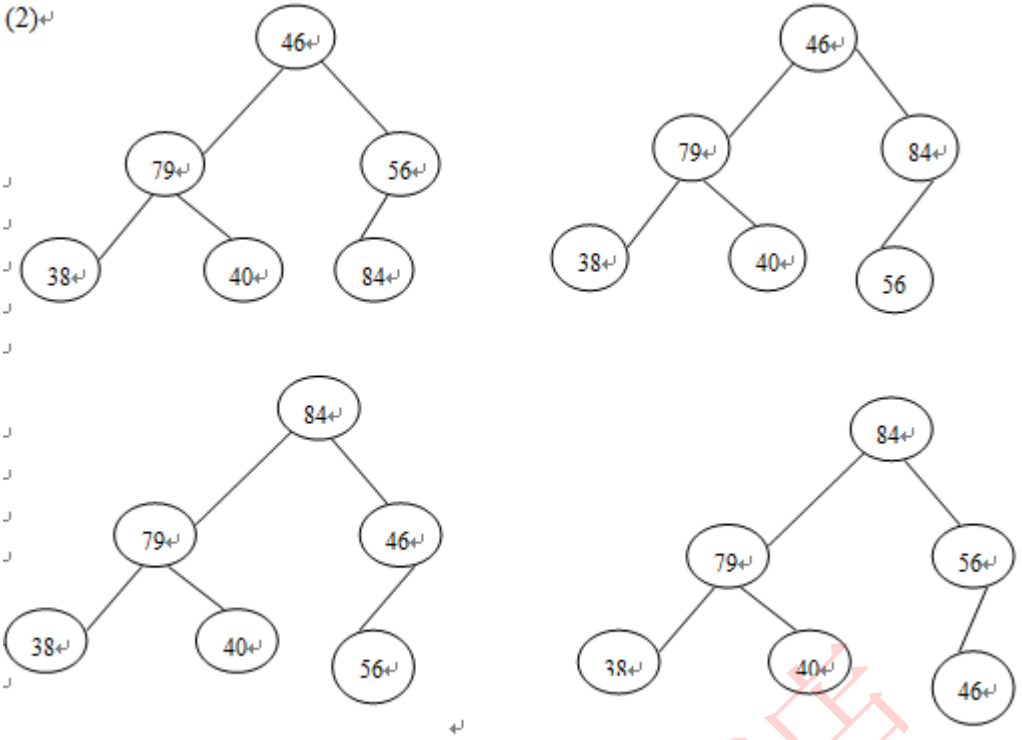


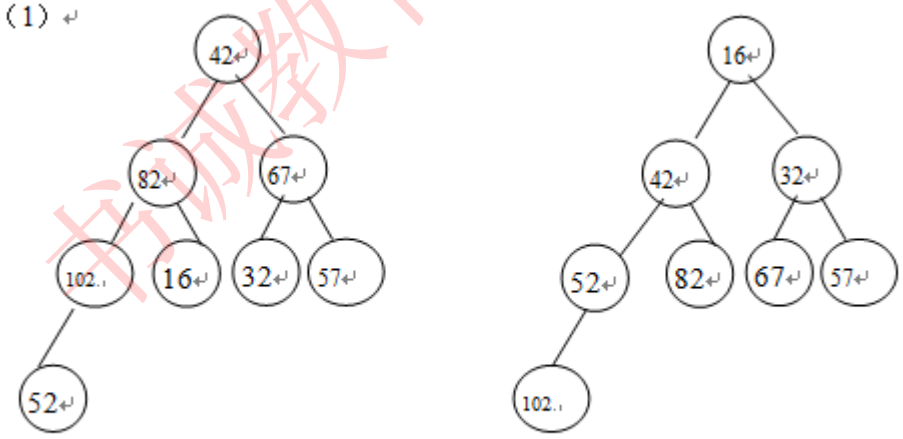
图 3

3．**综合题**（1）利用筛选过程把序列{42，82，67，102，16，32，57，52}建成堆（小根堆），画出相应的完全二叉树（不要求中间过程）

（2）写出对上述堆对应的完全二叉树进行中序遍历得到的序列

下面答案：

（1）



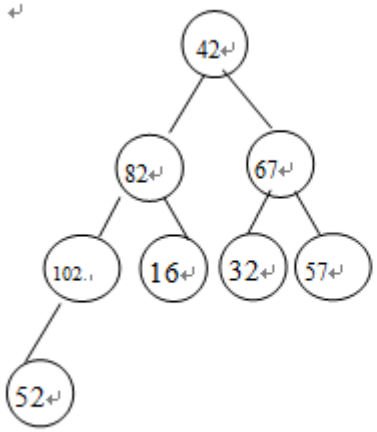
初始树　　堆

4．**综合题**（1）利用筛选过程把序列{42，82，67，102，16，32，57，52}建成堆（小根堆），画出相应的完全二叉树（不要求中间过程）

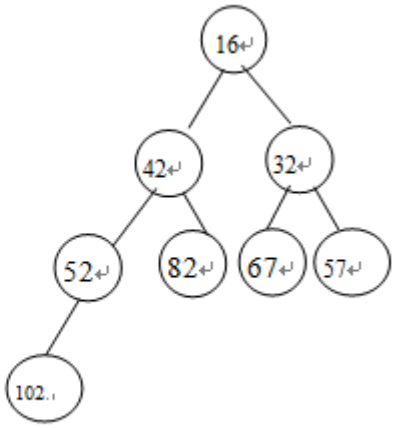
（2）写出对上述堆对应的完全二叉树进行中序遍历得到的序列

下面答案：

（1）



初始树



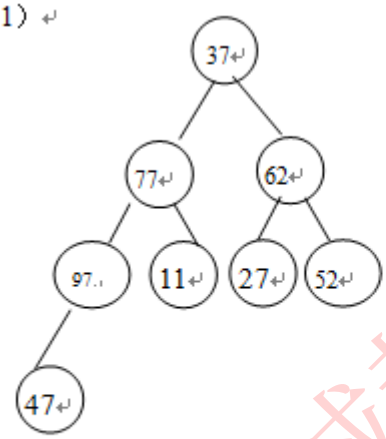
堆

（2）102，52，42，82，16，67，32，57

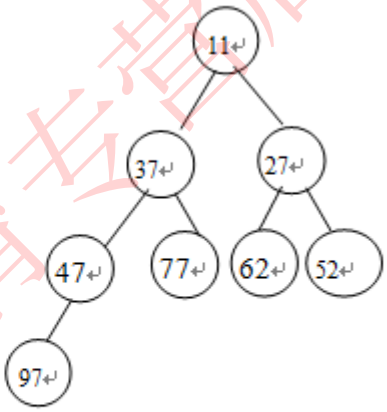
5．综合题（1）利用筛选法，把序列{37，77，62，97，11，27，52，47}建成堆（小根堆），画出相应的完全二叉树　　（2）写出对上述堆所对应的二叉树进行前序遍历得到的序列

下面答案：

（1）



初始树



堆

（2）11，37，47，97，77，27，62，52

9.8 选择排序之归并排序

9.9 排序稳定性题

1．按某关键字对记录序列排序，若关键字_____的记录在排序前和排序后仍保持它们的前后关系，则排序算法是稳定的，否则是不稳定的。　【关键字相等的记录】