计算机考研统考 408

第一部分 数据结构

计算机考研复习计划安排：

- 基础夯实阶段：时间一般从 3 月份开始到 7 月份左右。这一阶段选用的复习资料主要是和大纲比较吻合的教材以及配套的习题。
- 强化复习阶段：时间一般是 8 到 10 月份。这一阶段承接之前基础阶段的内容，进一步对重点内容，重点知识进行强化复习。
- 系统总结模拟训练的阶段: 时间一般从 11 月份到考前。这一阶段考生必须对学过的知识进行系统总结，找出自己的薄弱环节，查漏补缺，同时要精选一定量的模拟试题或历年真题演练。

数据结构考查目标

- 掌握数据结构的基本概念、基本原理和基本方法。
- 掌握数据的逻辑结构、存储结构及基本操作的实现，能够对算法进行基本的时间复杂度与空间复杂度的分析。
- 能够运用数据结构基本原理和方法进行问题的分析与求解，具备采用 C 或 C++语言设计与实现算法的能力。

知识框架

- 线性表
- 栈，队列，数组
- 树与二叉树
- 图
- 查找
- 排序

绪论

考点：算法和算法的衡量

1. 算法的定义

算法是对特定问题求解步骤的一种描述，是指令的有限序列。

2. 算法的时间复杂度

以基本运算的原操作重复执行的次数作为算法的时间度量。

常见的渐进时间复杂度有：O (1)＜O (log2n)＜O (n)＜O (nlog2n)＜O (n2)＜O (n3)＜O (2ⁿ) ＜0(n!) ＜0(nⁿ)。

3. 算法的空间复杂度

1. 以下算法的时间复杂度是（ ）。

```
void f (int n){
int x=1;
while (x<n)
x=2x;
}
```

A.O(log_2n)　　B. O(n)

C. O($nlog_2n$)　　D. O(n^2)

2. 以下算法的时间复杂度是（ ）。

```
void f (int n){
int p=1,d=n,f=n;
while (d>0){
if (d%2= =1) p=p*f;
f=f*f; d=d/2;
}
}
```

3.求出下列算法的时间复杂度。

1). 比较同一简单类型的两个数据 x1 和 x2 的大小，对于 x1>x2, x1=x2 和 x1<x2 这三种不同情况分别返回‘>’、 ‘=’和‘<’字符。

```
char compare(SimpleType x1,SimpleType x2)
{
    if (x1>x2) return‘>’;
    else if (x1==x2) return ‘=’;
    else return‘<’;
}
```

其时间复杂度为 O (1)

2). 将一个字符串中的所有字符按相反方向的次序重新放置。

```
void Reverse(char *p)
```

```
{
    int n=strlen(p);
    for (int i=0; i<n/2; i++) {
    char ch;
    ch=p[i]；
    p[i]=p[n-i-1];
    p[n-i-1]=ch;
    }
 }
```

其时间复杂度为 O (n)

3).从二维整型数组 a[m][n]中查找出最大元素所在的行、列下标。

```
void Find(int a[M][N],int m,int n,int &Lin,int &Col)
{ //M 和 N 为全局常量，应满足 M>=n 和 N>=n 的条件， Lin 和 Col 为引用形参，它是对应实参的别名，其值由实参带回 Lin=0; Col=0;
for (int i=0;i<m;i++)
    for (int j=0;j<n;j++)
      if (a[i][j]>a[Lin][Col]) {
          Lin=i; Col=j;
            }
}
```

其时间复杂度为 O (m*n)。

4.设线性表 L=（a1，a2，a3，…，an-2，an-1，an）采用带头结点的单链表保存，链表中结点定义如下：

```
typedef  struct  node
{ int data;
   struct node *next;
 }NODE;
```

请设计一个空间复杂度为 O(1)且时间上尽可能高效的算法，重新排列 L'=（a1，an，a2，an-1，a3，an-2，…）中的各结点，得到线性表 要求：

(1)给出算法的基本设计思想。

(2)根据设计思想，采用 C 或 C++语言描述算法，关键之处给出注释。

(3)说明你所设计的算法的时间复杂度。

参考答案 ：

(1)算法的基本设计思想：

算法分 3 步完成。第 1 步，采用两个指针交替前行，找到单链表的中间结点；第 2 步，将单链表的后半段结点原地逆置；第 3 步，从单链表前后两段中依次各取一个结点，按要求重排。

(2)算法实现：

```
void change_list( NODE * h)
{NODE *p, *q, *r, *s;
p=q=h;
while (q->next!=NULL)      //寻找中间结点
{p=p->next                 //p 走一步
 q=q->next;
 if(q->next!=NULL)q=q->next; //q 走两步
}
q=p->next;//p 所指结点为中间结点，q 为后半段链表的首结点
p->next=NULL;
while(q!=NULL)//将链表后半段逆置
{ r=q->next;
  q->next=p->next;
 p->next=q;
 q=r;
}
s=h->next;      //s 指向前半段的第一个数据结点，即插入点
q=p->next;      //q 指向后半段的第一个数据结点
p->next=NULL;
while(q!=NULL)//将链表后半段的结点插入到指定位置
{r=q->next;      //r 指向后半段的下一个结点
 q->next=s->next;//将 q 所指结点插入到 s 所指结点之后
 s->next=q;

 s=q->next;//s 指向前半段的下一个插入点
 q=r;
}
}
```

(3)算法的时间复杂度：

参考答案的时间复杂度为$O(n)$

第一章　线性表

大纲要求：

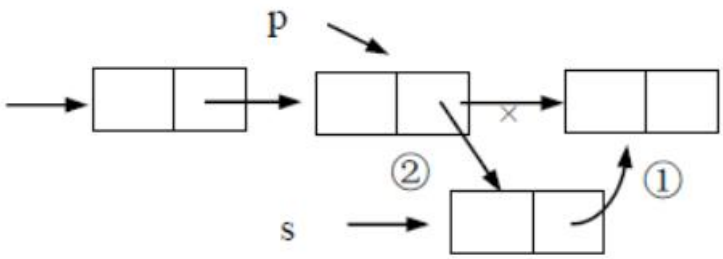
线性表的定义和基本操作（　　）

线性表的实现：顺序存储结构，链式存储结构。（　　）

线性表的应用（　　）

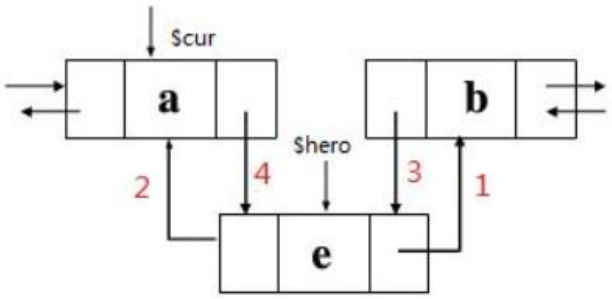
线性表基本知识

- 线性表：顺序表，链表
- 顺序表：初始化，插入运算，删除运算
- 链表：单链表实现，插入操作，删除操作，循环链表，双向链表



在*p之后插入*s

插入操作



删除操作

一、基本知识

1.以下属于逻辑结构的是(　)。

A.顺序表　　B.哈希表

C.有序表　　D.单链表

2.线性表是具有 n 个(　)的有限序列(n>0)。

A.表元素　　B.字符

C.数据元素　　D.数据项

3.对于一个线性表既要求能够进行较快的插入和删除,又要求存储结构能反映数据之间的逻辑关系,则应该用(　)。

A.顺序存储方式

B.链式存储方式

C.散列存储方式

D.以上均可以

二、顺序表的存储和操作

题型分析:这类题型主要考查考生对顺序表的本质的理解。顺序表本质上是线性表的顺序存储。也就是说,元素一个挨着一个地存放在一段连续的内存空间中。

4.一个顺序表第一个数据元素的地址是100,每个数据元素的长度为2,则第五个数据元素的地址是(　)。

A.108　　B.110

C.100　　D.120

5.向一个长度为n的顺序表中的第i个元素(0<i<n-1)之前插入一个元素时,需要向后移动()个元素。

A.n　　B.n-i

C.n-i+1　　D.n-i-1

三、动态链表的操作以及结点

题型分析:要明白动态链表的实质。一方面它像火车车厢一样,一节接着一节;另一方面,它又和火车不一样,因为“接着”是通过“指针”来实现的。所以,对指针的深刻理解也是解决这类问题的关键。

6.在一个单链表中,若p所指结点不是最后结点,在p之后插入s所指结点,则执行(　)。

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

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

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

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

7.在一个以h为头的单循环链表中,p指针指向链尾的条件是(　)。

A.p->next->next=h

B.p->next=h

C.p->next=h->next

D.p->next->NULL

8.在一个单链表中,若删除p所指结点的后继结点,则执行(　)。

A.p=p->next->next;

B.p=p->next;

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

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

9. 已知一个带头结点的单链表,假如该链表只给出了头指针 L,在不改变链表的前提下设计一个算法:查找链表中“倒数”第 k(k 为正整数)个结点,若查找成功,输出该结点的 data 值,并返回 1;否则,返回 0。链表结点结构如下:

要求:

(1)描述算法的基本设计思想。

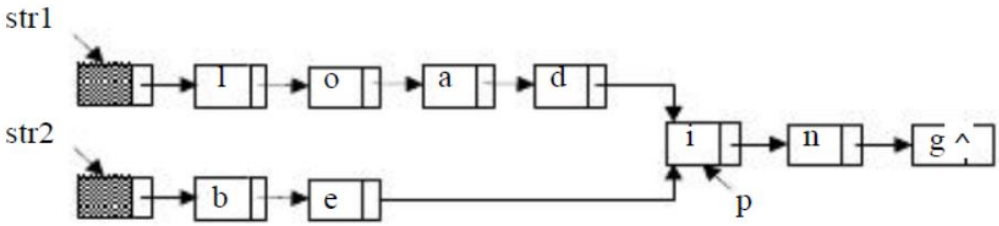
(2) 描述算法的详细实现步骤。

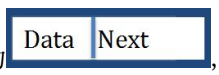
(3)根据设计思想和实现步骤,用程序设计语言描述算法(使用 C 或 C++或 Java 语言实现),关键之处请给出简要注释。

算法描述:

```
int  LocateElement(linklist list, int k){
    P1=list->next;
    P=list;
    i=1;
    while(P1) {
      P1=p1->next;
      i++;
      if(i > k)  p=p->next;          //如果 i>k , 则 P 也往后移动
    }
    if(p==list)  return 0;           //说明链表就没有 k 个结点
    else {
      printf("%d\n",p->data);
      return 1;
    }
}
```

10. 假定采用带头结点的单链表保存单词,当两个单词有相同的后缀时,则可以共享相同的后缀存储空间。例如,"loading"和"being"的存储映像如图所示。



设 str1 和 str2 分别指向两个单词所在单链表的头结点,链表结点结构为 

,请设计一个时间上尽可能高效的算法,找出由 str1 和 str2 所指的两个链表共同后缀的起始位置(如图中字符 1 所在结点的位置 p)。

要求:

(1) 给出算法的基本设计思想。

(2)根据设计思想,用 C 或 C++或 Java 语言描述算法,关键之处给出注释。

(3)说明你所设计算法的时间复杂度。

算法实现：

```
typedef  struct Node{
    char  data;
    struct Node *next;
}SNODE;
SNODE *findList(SNODE *str1, SNODE *str2){
  int m,n;
  SNODE *p,*q;
  m=listLen(str1);    //求 str1 的长度。O(m)
  n=listLen(str2);    //求 str1 的长度。O(m)
  //下面两个循环是让 p、q 所指向的链表等长，下面三个循环的时间复杂度 O(max(m,n))
  for(p=str1;m>n;m--)  p=p->next;
  for(q=str2;m<n;n--)  q=q->next;
  while(p->next !=NULL && p->next !=q->next){  //查找共同后缀的起点
     p=p->next;
     q=q->next;
  }
  return  p->next;
}
int listLen(SNODE* head){    //求表长
  int len=0;
  while(head->next !=NULL){
    len++;
    head=head->next;
  }
  return len;
}
```

四、双向链表

题型分析:关于双向链表,要掌握三方面内容:第一,双向链表中的每个结点有两个指针域;第二,从双向链表中的任意一个结点出发,均可以到达其他各个结点;第三,双向链表中结点的插入和删除一般要修改 4 个指针(有些场合要注意修改指针的顺序)。

11.对于双向链表,在两个结点之间插入一个新结点需修改的指针为（ ）个，单链表为( )个。

A.2,4　　B.2,2

C.4,2　　D.4,4

12.在双向链表指针 p 的结点前插入一个指针 q 的结点操作是( )。

A.p->llink=q;q->rlink=p;p->llink->rlink=q;q->llink-q;

B.p->llink=q;p->llink->rlink=q;q->rlink-p;q->llink=p->llink;

C.q->rlink=p;q->llink=p->llink;p->llink->rlink=q;

D.p->llink=q;q->rlink=q;p->llink-q;p->llink=q;

13.在双向链表中删除指针 p 所指的结点时,需要修改的指针是( )。

A.p->llink->rlink-p->rlink;p->rlink->llink-p->llink;

B.p->llink-p->llink->llink;p->llink->rlink=p;

C.p->rlink->llink=p;p->rlink=p->rlink->rlink

D.p->rlink-p->llink->llink;p->llink=p->rlink->rlink;

五、静态链表

题型分析:静态链表的本质是链表,只不过指针是借助于数组下标来实现的,当然它需要连续的内存空间。

14.静态链表中指针表示的是(　)。

A.下一个元素的地址

B.内存储器的地址

C.下一个元素在数组中的位置

D.右链或左链指向的元素的地址

15.以下说法中错误的是(　)。

①.静态链表既有顺序表的优点,又有动态链表的优点。所以它存取表中第 1 个元素的时间与 i 无关

②.静态链表中能容纳的元素的个数在定义时就确定了,以后不能增加

③.静态链表与动态链表的插入、删除类似,不需要移动元素。

A.①　　B.①,②

C.①,②,③　　D.②,③

六、插入和删除结点的时间复杂度

题型分析:线性表的结点插入和删除操作,一个重要的问题就是寻找位置,这很费时间。另一个重要的问题就是顺序表在插入和删除时会牵涉大量元素的移动,所以这两点是求时间复杂度的突破口。

16.在有 n 个结点的线性表的数组实现中,算法的时间复杂度为 O(1)的操作是(　)。

A.访问第 i 个结点(1<i<n)和求第 i 个结点的直接前驱(2<i<n)

B.在第 i 个结点后插入一个新的结点(1<i<n)

C.删除第 i 个结点

D.以上都不对

17.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的算法的时间复杂度为(　)(1<i<n+1)。

A.O(0)　　B.O(1)

C.O(n)　　D.$O(n^2)$

18.线性表(a1,a2,…,an)以链接方式存储时,访问第 i 个元素的时间复杂度为(　)。

A.O(i)　　B.O(1)

C.O(n)　　D.O(i-1)

真题强化

1.下列函数的时间复杂度是（　　）.

```
int func（int n）{
        int i=0,sum=0;
        while(sum<n)
                sum+=++i;
        return i;
    }
```

A. $O(\log_2 n)$　　B. $O(n^{1/2})$　　C. $O(n)$　　DO $(n\log_2 n)$

2. 下列程序段的时间复杂度是（　　）.

```
count=0;
for(k=1;k<=n;k*=2)
        for(j=1;j<=n;j++)
                count++ ;
```

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

3. 下列数据结构中，（　　）是非线性数据结构.

- A.栈
- B.队列
- C.二叉树
- D.堆

4.链表不具有的特点是（　）

A.插入、删除操作不需要移动元素

B.可随机访问任一元素

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

5.已知一个带有表头节点的双向循环链表 L，节点结构为

其中，prev 和 next 分别是指向其直接前去和直接后继节点的指针。现要删除指针 p 所指的节点，正确的语句序列是（　）.

A.p->next->prev=p->prev;p->prev->next=p->prev;free(p);

B.p->next->prev=p->next;p->prev->next=p->next;free(p);

C.p->next->prev=p->next;p->prev->next=p->prev;free(p);

D.p->next->prev=p->prev;p->prev->next=p->next;free(p);

6.在包含 1000 个元素的线性表中实现如下各运算，所需执行时间最长的是（　）.

A.线性表按顺序方式存储，删除线性表的第 900 个节点

B.线性表按链式方式存储，删除指针 p 所指向的节点

C.线性表按顺序方式存储，在线性表的第 100 个节点后面插入一个新节点

D.线性表按链式方式存储，在线性表的第 100 个节点后面插入一个新节点

7.在顺序表（长度为 127）中插入.一个元素平均要移动（　）个元素.

A.8　　B.63.5

C.63　　D.7

8.某线性表中，最常用的操作是在最后一个元素之后插入一个元素和删除一个元素，则采用（　）存储方式最节省时间.

A.单链表

B.仅有头指针的单循环链表

C.双链表

D.仅有尾指针的单循环链表

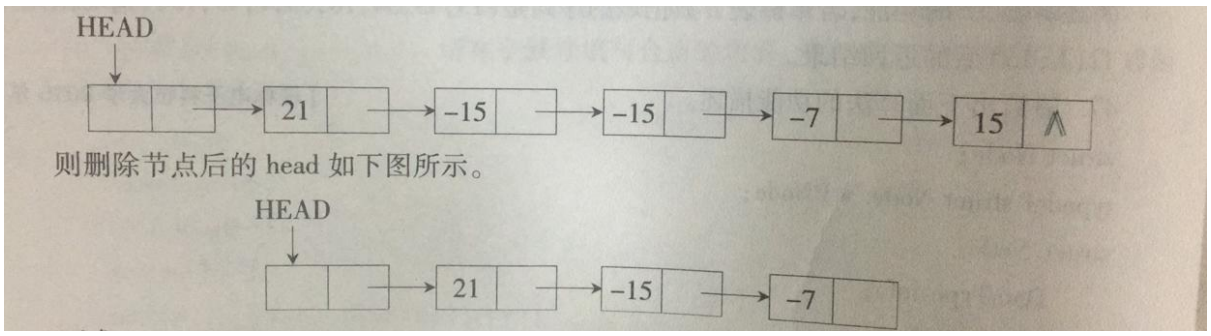
9.设有两个集合 A 和集合 B，设计生成集合 C=A∩B 的算法，其中集合 A、B 和 C 用数组存储表示。

（1）给出算法的基本设计思想。

（2）根据设计思想，采用 C 或 C++或 Java 语言表述算法，关键之处给出注释。

（3）说明你所设计算法的时间复杂度。

10.用单链表保存 m 个证书，节点的结构为[data][link]，且|data|≤n（n 为正整数）。现要求设计一个时间复杂度尽可能高效的算法，对于链表中 data 的绝对值相等的节点，仅保留第一次出现的节点而删除其余绝对值相等的节点。例如，若给定的单链表 head 如下图所示。



要求：

（1）给出算法的基本设计思想。

（2）C 或 C++语言，给出链表节点的数据类型定义。

（3）根据设计思想，采用 C 或 C++语言描述算法，关键之处给出注释。

（4）说明你所设计算法的时间复杂度和空间复杂度。

更多典型题目

1.若某线性表中最常用的操作是在最后一个结点之后插入一个结点和删除第一个结点，则下面最节省运算时间的存储方式是（　）。

A.单链表

B.带有头指针的单循环链表

C.双链表

D.带有尾指针的单循环链表

2.已知两个长度分别为 l 和 s 的降序链表，若将它们合并为一个长度为 l+s 的升序链表，则最坏情况下的时间复杂度是（　）。

A.O（l）

B.O（ls）

C.O（min（l，s））

D.O（max（l，s））

3.线性表中存放的主要是（　）。

A.整型常量

B.字符

C 数据元素

D.信息元素

4.下面的叙述中正确的是（　）。

Ⅰ.线性表在链式存储时，查找第 i 个元素的时间同 i 的值成正比

Ⅱ.线性表在链式存储时，查找第 i 个元素的时间同 i 的值无关

Ⅲ.线性表在顺序存储时，查找第 i 个元素的时间同 i 的值成正比

A.仅Ⅰ　　B.仅Ⅱ

C.仅Ⅲ　　D.Ⅰ、Ⅱ、Ⅲ

5.对于某线性表来说，主要的操作是存取任一指定序号的元素和在最后进行插入运算，那么应该选择（　）存储方式最节省时间。

A.顺序表

B.双链表

C.带头结点的双循环链表

D.单循环链表

6.若线性表最常用的运算是查找第 i 个元素及其前驱的值，则下列存储方式中最节省时间的是（　）

A.单链表　　B.双链表

C.单循环链表　　D.顺序表

7.如果线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用（　）存储方式最节省运算时间。

A.单链表

B.仅有头指针的单循环链表

C.双链表

D.仅有尾指针的单循环链表

8.算法的时间复杂度取决于（　）。

A.问题的规模

B.待处理数据的初态

C.A 和 B

D.以上都不正确

9.关于链表的特点，下面的叙述中不正确的是（　）。

A.插入、删除运算方便

B.可实现随机访问任一元素

C.不必事先估计存储空间

D.所需空间与线性长度成正比

10.设线性表中有 2n 个元素，以下操作中，在单链表上实现要比在顺序表上实现效率更高的是（　）。

A.删除指定元素

B.在最后一个元素的后面插入一个新元素

C.顺序输出前 k 个元素

D.交换第 i 个元素和第 2n-i-1 个元素的值（i=0，1，…，n-1）

11.下面的算法实现的是带附加头结点的单链表数据结点逆序连接，空缺处应当填入（　）

```
viod reverse(pointer h)      //h 为附加头结点指针
      pointer p,q;
```

```
p=h->next; h->next=NULL;
while(p != null){
    q=p;
    p=p->next;
    q->next=h->next;
    h->next=(____);
}
}
```

A. h

B. p

C. q

D. q-> next

12. 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的算法的时间复杂度为（ ）（1≤i≤n+1）。

A. O（0）　　B. O（1）

C. O（n）　　D. O（n2）

13. 线性表（a1，a2，…，an）以链式存储方式存储时，访问第 i 位置元素的时间复杂度为（ ）。

A. O（i）　　B. O（1）

C. O（n）　　D. O（i-1）

14. 非空的循环单链表 head 的尾结点 p 满足（ ）。

A. p-> next=head

B. p-> next=NULL

C. p=NULL

D. p=head

15. 静态链表中指针表示的是（ ）。

A. 内存地址

B. 数组下标

C. 下一元素数组下标

D. 左、右孩子地址

16. 在单链表指针为 p 的结点之后插入指针为 s 的结点，正确的操作是（ ）。

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

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

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

D.p-＆gt；next=s-＆gt；next；p-＆gt；next=s；

17.对于一个头指针为head的带头结点的单链表，判定该表为空表的条件是（　）。

A.head==NULL

B.head-＆gt；next==NULL

C.head-＆gt；next==head

D.head！=NULL

18.以下与数据的存储结构无关的术语是（　）。

A.循环队列

B.链表

C.哈希表

D.栈

19.以下数据结构中，（　）是线性数据结构。

A.广义表

B.二叉树

C.稀疏矩阵

D.串

综合题

1.有两个集合A和B，利用带头结点链表表示，设头指针分别为la和lb。两集合的链表元素皆为递增有序。设计一个算法，将A与B合并，合并后仍然保持整个链表中的数据依次递增。不得利用额外的结点空间，只能在A和B的原有结点空间上完成。要求：

（1）给出算法的基本设计思想。

（2）根据设计思想，采用C或C++或Java语言描述算法，关键之处给出注释。

（3）分别给出算法各部分的时间复杂度。

2.线性表（a1，a2，a3，…，an）中元素递增有序且按顺序存储于计算机内。要求设计一算法用最少时间在表中查找数值为x的元素，并将其与后继元素位置相交换。如果线性表中找不到该元素，则将该元素插入表中并使表中元素仍递增有序。

（1）给出算法的基本设计思想。

（2）根据设计思想，采用C或C++或Java语言描述算法，关键之处给出注释。

（3）分别给出算法各部分的时间复杂度。

3.已知顺序表A，在不改变顺序表中奇数号元素与偶数号元素相对位置的前提下，设计算法，将所有奇数号元素移到所有偶数号元素前。

（1）给出算法的基本设计思想。

（2）根据设计思想，采用C或C++或Java语言描述算法，关键之处给出注释。

4.已知单链表L是一个递增有序表，试写一高效算法，删除表中值大于min且小于max的结点（若表中有这样的结点），同时释放被删结点的空间，这里min和max是两个给定的参数。

5.线性表（a1，a2，a3，…，an）中元素递增有序且按顺序存储于计算机内。要求设计算法完成以下内容：
（1）用最少的时间在表中查找数值为 x 的元素。
（2）若找到将其与后继元素位置相交换。
（3）若找不到将其插入表中并使表中元素仍递增有序。

6.设有一个双链表 L，每个结点中除有 prior、data 和 next 这 3 个域外，还有一个访问频度域 freq，在链表被启用之前，其值均初始化为零。每当在链表进行一次 LocateNode（L，x）运算时，令元素值为 x 的结点中 freq 域的值加 1，并调整表中结点的次序，使其按访问频度的递减排列，以便使频繁访问的结点总是靠近表头。试写一符合上述要求的 LocateNode 运算的算法。

7.有一个不带头结点的单链表 list，链表中结点都有两个域：数据域 data 和指针域 link。已知初始时该单链表无序，请设计一个算法将该链表按结点数据域的值的大小，将其从小到大依次重新链接，在链接过程中不得使用除该链表以外的任何链结点空间。要求：
（1）给出算法的基本设计思想。
（2）根据设计思想，采用 C 或 C++或 Java 语言描述算法，关键之处给出注释。

8.如果以单链表表示集合，设集合 A 用单链表 LA 表示，集合 B 用单链表 LB 表示，设计算法求两个集合的差，即 A-B。

9.已知 3 个带头结点的线性链表 A、B、C 中的结点均依元素值自小至大非递减排列（可能存在两个以上值相同的结点），编写算法对链表 A 进行如下操作：使操作后的链表 A 中仅留下 3 个表中均包含的数据元素的结点，且没有值相同的结点，并释放所有无用结点。限定算法的时间复杂度为 O（m+n+p），其中 m、n 和 p 分别为 3 个表的长度。

10.已知非空链表 A，其指针是 list，链表中的结点由两部分组成：数据域 data 和指针域 link。设计一个算法，将链表中数据域值最小的那个链结点移到链表的最前面，在不额外申请新的链结点的情况下，使得算法时间复杂度和空间复杂度尽可能低。要求：
（1）给出算法的基本设计思想。
（2）根据设计思想，采用 C 或 C++或 Java 语言描述算法，关键之处给出注释。

11.已知一个双向链表，其结点结构为数据域 data、左指针域 llink、右指针域 rlink；设指针 P 指向双向链表中的某个结点。写出一个算法，实现 P 所指向的结点和它的前缀结点之间顺序的互换。要求：
（1）给出算法的基本设计思想。
（2）根据设计思想，采用 C 或 C++或 Java 语言描述算法，关键之处给出注释。

12.两个整数序列 A=a1，a2，a3，…，am 和 B=b1，b2，b3，…，bn 已经存入两个单链表中，设计一个算法，判断序列 B 是否是序列 A 的子序列。

13.已知一个带有头结点的单链表 L，其结点结构由两部分组成：数据域 data，指针域 link。设计一个算法，以最高效的方法实现在单链表中删除数据域最小值结点。要求：
（1）给出算法的基本设计思想。

（2）根据设计思想，采用 C 或 C++或 Java 语言描述算法，关键之处给出注释。

14.设有集合 A 和集合 B，要求设计生成集合 C=A∩B 的算法，其中集合 A、集合 B 和集合 C 用链式存储结构表示。

第二章　栈队列数组

栈队列数组大纲

- 栈、队列和数组
- 栈和队列的基本概念（☆☆☆）
- 栈和队列的存储结构（☆☆）
- 栈和队列的应用（☆☆）
- 特殊矩阵的压缩存储（☆）

基本知识

- 栈的定义，基本操作：入栈，出栈。
- 栈的常见应用
- 队列的定义，基本操作：插入，删除。
- 其他形式的队列
- 特殊矩阵
-

一、出栈和出队的顺序问题

题型分析:本题型主要考查栈和队列的特性,只要始终清楚元素如何入栈和入队,问题就解决了对于栈,只能在栈顶进,栈顶出;对于队列,只能在队尾入,队头出。

1.设输入序列 1,2,3,4,则下述序列中(　)不可能是出栈序列。

A.1,2,3,4

B.4,3,2,1

C.1,3,4,2

D.4,1,2,3

2.已知一个栈的进栈序列是 1,2,3,..,,其输出序列是 P1,P2,P3,…,Pn,若 P1=3 则 P2 为(　)。

A.可能是 2

B.一定是 2

C.可能是 1

D.一定是 1

3.某队列允许在其两端进行入队操作,但仅允许在一端进行出队操作,若元素 a,b,c,d,e 依次入此队列后再进行出队操作,则不可能得到的序列是(　)。

A.bacde

B.dbace

C.dbcae

D.ecbad

4.元素 a,b,c,d,e 依次进入初始为空的栈中,若元素进栈后可停留,可出栈,直到所有,元素都出栈,则在所有可能的出栈序列中以元素 d 开头的序列个数是(　)。

A.3　　B.4　　C.5　　D.6

二、循环队列的问题

题型分析:循环队列其实就是队列顺序存储结构的一种特例,它是在解决假溢出时提出的。

5.已知循环队列存储在一维数组A[0..n-1]中,且队列非空时front和rear分别指向队头元素和队尾元素。若初始时队列为空,且要求第一个进入队列的元素存储在A[0]处,则初始时front和rear的值分别是( )。

A.0,0

B.0,n-1

C.n-1,0

D.n-1,n-1

6.假设以数组A[m]存放循环队列的元素,其头、尾指针分别为front和rear,则当前队列中的元素个数为( )。

A.(rear-front)%m

B.(front-rear+m)%m

C.(rear-front+m)%m

D.rear-front+1

7.若用一个大小为6的数组来实现循环队列,且当前rear和front的值分别为0和3,当从队列中删除一个元素,再加上两个元素后,则rear和front的值分别为( )。

A.1和5

B.4和2

C.2和4

D.5和2

三、多维数组的存储问题

题型分析:此类题型主要考查考生对多维数组存储本质的掌握。解决此类题分为两步:第一步,弄清存储规则,到底是低下标优先还是高下标优先;第二步,按照这个规则按部就班地存储。

8.二维数组A按行顺序存储,其中每个元素占1个存储单元。若A[1][1]的存储地址为420,A[3][3]的存储地址为446,则A[5][5]的存储地址为( )。

A.472　　B.471

C.2和4　　D.5和2

9.设一个10阶的对称矩阵A采用压缩存储方式,以行为主序存储,a11为第一个元素,其存位地址为1,每个元素占一个地址空间,则a85的地址为( ).

A.13　　B.33　　C.32　　D.40

四、栈和队列的空间需求问题

题型分析:此类题型的解决方案是,手工模拟计算机的执行过程,分析在执行的过程中,栈或队列中最多有多少个元素,元素数就是栈和队列的最小需要空间。

10.设栈S和队列Q的初始状态均为空,元素abcdefg依次进入栈S。若每个元素出栈后立即

进入队列 Q,且 7 个元素出队的顺序是 bdcfeag,则栈 S 的容量至少是(　)。

A.1　　B.2　　C.3　　D.4

11.若栈采用顺序存储方式存储,现在两栈共享同一空间 v[1..m],top[i]代表第 i 个栈(i=1,2)的栈顶元素的下标,栈 1 的底在 v[1]的位置,栈 2 的底在 v[m]的位置,则栈满的条件是(　)。

A.top[2]-top[1]==0

B.top[1]+1==top[2]

C.top[1]+top[2]==m

D.top[1]==top[2]

12.已知操作符+，-，*，/(和)。将中缀表达式 a+b-a*((c+d)/e-f)+g 转换为等价的后缀表达式 ab+acd+e/f-*-g+时,用栈来存放暂时还不能确定运算次序的操作符。若栈的初始时为空,则转换过程中同时保存在栈中的操作符的最大个数是(　)。

A.5　　B.7　　C.8　　D.11

五、递归和非递归的转换

题型分析:要想解决这类问题,必须清楚递归程序的执行过程,之后,用栈来实现函数调用时自现场保护与恢复问题。

13.将下列递归过程改为非递归过程。

```
void  test( int  &sum)
    int  x;
    scanf(x);
    if(x==0)  sum=0;
    else
        test(sum);
        sum+=x;
    }
    printf(sum);
}
```

```
void  test(int  &sum)  {
    Stack  S;
    Int x;
    scanf(x);
    InitStack(S);
    while(x){
        Push(S,x);
        scanf(x);
    }
    sum=0;
    printf(sum);
    while(Pop(S,x)){
        sum+=x;
        printf(sum);
    }
}
```

六、栈和队列的应用

14. 利用两个栈 S1、S2 来模拟一个队列。已知栈的三种运算定义如下:

Push(ST,x):元素 x 入栈 ST;

Pop(ST,x):ST 栈顶元素出栈,赋给变量 x;

Sempty(ST):判断 ST 栈是否为空。

利用栈的这三种运算来实现该队列的三种运算:

Enqueue:插入一个元素入队列;

Dequeue:删除一个元素出队列;

Queueempty:判队列是否为空。

入队

```
status  Enqueue(Stack  S1,  Elemtype  e){
  //S1 是容量为 n 的栈,现将元素 e 入栈,成功返回 OK,否则返回 ERROR
  if(top1 ==n && ! Sempty(S2)){
      printf("栈满");
      return  ERROR;
  }
  if(top1 ==n && Sempty(S2)){
      while(! Sempty(S1)){
          Pop(S1, e);
          Push(S2, e);
      }
  }
  Push(S1, e);
  return  OK;
}
```

出队

```
  void Dequeue(Stack S1, Stack S2){
      if(! Sempty(S2){
          Pop(S2, e);
          Printf("出队元素为",e);
      }
      else{
          if(Sempty(S1)){
              printf("队列空");
              return ERROR;
          }
          else{
              while(! Sempty(S1)){
                  Pop(S1,e);
                  Push(S2,e)
              }
              Pop(S2, e);
              printf("出队元素",e);
          }
      }
  }
```

判空

```
bool  Queue_empty(){
    if(Sempty(S1)&&Sempty(S2))
        return true;
    else
        return false;
}
```

真题

1.下列关于栈的叙述中，错误的是（ ）.

Ⅰ.采用非递归方式重写递归程序时，必须使用栈

Ⅱ.函数调用时，系统要用栈保存必要的信息

Ⅲ.只要确定了入栈次序，即可确定出栈次序

Ⅳ.栈是一种受限的线性表，允许在其两端进行操作

A.仅Ⅰ　　B.Ⅰ、Ⅱ、Ⅲ

C.Ⅰ、Ⅲ、Ⅳ　　D.Ⅱ、Ⅲ、Ⅳ

2.已知程序如下：

```
intS（intn）{
    return（n<=0）?0:S(n－1)＋n;
}
voidmain（）{
    cout<<S（1）;
}
```

程序运行中使用栈来保存调用过程的信息，自栈底到栈顶保存的信息一次对应的是（）。

A.main（）→S（1）→S（0）　　B.S（0）→S（1）→main（）

C.main（）→S（0）→S（1）　　D.S（1）→S（0）→main（）

3.一个栈的入栈序列为1，2，3，•••，n，其出栈序列是P1,P2,P3，•••，Pn。若P2=3，则P3可能取值的个数是（）.

A.n－3　　B.n－2　　C.n－1　　D.无法确定

4.若元素a，b，c，d，e，f依次进栈，允许进栈、退栈操作交替进行，但不允许连续三次进行退栈操作，则不可能得到的出栈序列是（ ）。

A.d，c，e，b，f，a

B.c，b，d，a，e，f

C.b，c，a，e，f，d

D.a，f，e，d，c，b

5.为解决计算机主机与打印机之间速度不匹配问题，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，而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是（　）。

A.栈　　B.队列　　C.树　　D.图

6.若一个栈以向量V[1•••n]存储，初始栈顶指针top为n＋1，则下面x入栈的正确操作是（）。

A.top＝top＋1；V[top]＝x；

B.V[top]＝x；top＝top＋1；

C.top＝top－1；V[top]＝x；

D.V[top]＝x；top＝top－1；

7.一个栈的输入序列未1，2，3，•••，n，若输出序列的第一个元素是n，输出第i（1≤i≤n）个元素是（ ）.

A.n－i

B.n－i－1

C.n－i＋1

D.i

8.若用一个大小为6的数组来实现循环队列，且当前rear和front的值分别为0和3.当从队列中删除一个元素，再加上两个元素后，rear和front的值分别为多少？（ ）.

A.1和5

B.2和4

C.4和2

D.5和1

9.一个栈的输入序列为1，2，3，4，5，则下列序列中不可能是栈的输出序列的是（ ）.

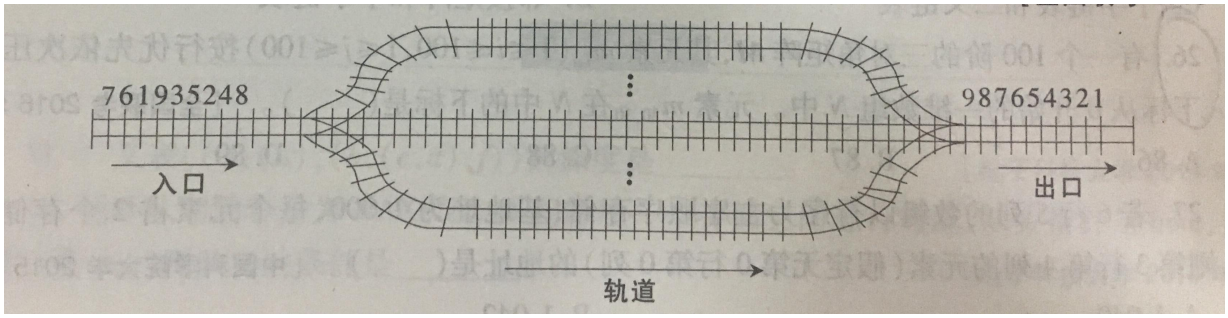
A.2,3,4,1,5

B.5,4,1,3,2

C.2,3,1,4,5

D.1,5,4,3,2

10.设有如下如所示的火车车轨，入口到出口之间有n条轨道，列车的行进方向均为从左到右，列车可驶入任意一条轨道。现有编号为1-9的9列列车，驶入的次序依次是8，4，2，5，3，9，1，6，7.若期望驶出的次序依次为1-9，则n至少有（ ）.



A.2　　B.3　　C.4　　D.5

11.循环队列存放在一组数组 A[0 ••• M－1]中，end1 指向队头元素，end2 指向队尾元素的后一个位置。假设队列两端均可进行入队和出队操作，队列中最多能容纳 M－1 个元素，初始时为空。下列判断队空和队满的条件中，正确的是（　）.

A.队空：end1==end2　　队满：end1==（end2＋1）modM

B.队空：end1==end2　　队满：end2==（end1＋1）mod（M－1）

C.队空：end2==（end1＋1）modM　　队满：end1==（end2＋1）modM

D.队空：end1==（end1＋1）modM　　队满：end2==（end1＋1）mod（M－1）

12.已知循环队列存储在一维数组 A[0 •••n－1]中，且队列非空时，front 和 rear 分别指向队头元素和队尾元素。若初始时队列为空，且要求第 1 个进入队列的元素存储在 A[0]处，则初始时 front 和 rear 的值分别为（　）.

A.0，0

B.0，n－1

C.n－1，0

D.n－1，n－1

13.最大容量为 n 的循环队列，队尾指针是 rear，队头指针是 front，则队满的条件是（ ）

A.（rear＋1）modn= = front

B.rear= = front

C.rear＋1= = front

D.（rear－1）modn= = front

14.适用于压缩存储稀疏矩阵的两种存储结构是（）.

A.三元组表和十字链表

B.三元组表和邻接矩阵

C.十字链表和二叉链表

D.邻接矩阵和十字链表

14.有一个 100 阶的三对角矩阵 M，其元素 mi，j（1≤i≤100；1≤j≤100）按行优先依次压缩存入下标从 0 开始的一维数组 N 中。元素 m30，30 在 N 中的下标是（ ）.

A.86　　B.87　　C.88　　D.89

15.下列关于数组的描述，正确的是（）.

A.数组的大小是固定的，但可以有不同的类型的数组元素

B.数组的大小是可变的，所有数组元素的类型必须相同

C.数组的大小是固定的，所有数组元素的类型必须相同

D.数组的大小是可变的，可以有不同的类型的数组元素

更多典型题目

1.栈和队列的主要区别在于（　）。

A.它们的逻辑结构不一样

B.它们的存储结构不一样

C.所包含的运算不一样

D.插入和删除运算的限定不一样

2.若循环队列以数组 Q［0..m-1］作为其存储结构，变量 rear 表示循环队列中的队尾元素的实际位置，其移动按 rear=（rear+1）MODm 进行，变量 length 表示当前循环队列中的元素个数，则循环队列的队首元素的实际位置是（　）。

A.rear-length

B.（rear-length+m）MODm

C.（rear-length+1+m）MODm

D.m-length

3.一个以向量 V［n］存储的栈，其初始栈顶指针 top 为 n+1，则对于 x，其正确的进栈操作是（　）。

A.top=top+1；V［top］=x

B.V［top］=x；top=top+1

C.top=top-1；V［top］=x

D.V［top］=x；top=top-1

4.为了增加内存空间的利用率和减少溢出的可能性，两个栈可以共享一片连续的内存空间，此时应将两栈的栈底分别设在（　）。

A.内存空间的首地址

B.内存空间的尾地址

C.内存空间的两端

D.内存空间的中间

5.已知输入序列为 abcd，经过输出受限的双端队列后，能得到的输出序列是（　）。

A.dacb

B.cadb

C.dbca

D.以上答案都不对

6.假设一个序列 1，2，3，…，n 依次进栈，如果出栈的第一个元素是 n，那么第 i（1≤i≤n）个出栈的元素是（　）。

A.不确定　　B.n-i+1

C.i　　D.n-i

7.假设一个序列 1，2，3，…，n 依次进栈，如果第一个出栈的元素是 i，那么第 j 个出栈的元素是（　）。

A.i-j-1　　B.i-j

C.j-i+1　　D.不确定的

8.已知当前栈中有 n 个元素，此时如果有新的元素需要执行进栈操作，但发生上溢，则由此可以判断，此栈的最大容量为（　）。

A.n-1　　B.n

C.n+1　　D.n/2

9.设有 5 个元素 a，b，c，d，e 顺序进栈，下列几个选项中，不可能的出栈序列是（　）。

A.a，b，c，d，e

B.d，e，c，b，a

C.a，c，e，b，d

D.c，b，a，d，e

10.有 6 个元素按 6，5，4，3，2，1 的顺序依次进栈，不合法的出栈序列是（　）。

A.543612　　B.453126

C.346521　　D.234156

11.有 5 个元素，其入栈次序为 A，B，C，D，E，在各种可能的出栈次序中，以元素 C，D 最先出栈的次序不包括（　）。

A.CDEBA　　B.CDBEA

C.CDBAE　　D.CDAEB

12.对于 4 个元素依次进栈，可以得到（　）种出栈序列。

A.10　　B.12　　C.14　　D.16

13.现有两栈，其共享空间为 V［1..m］，top［i］代表第 i 个栈（i=1，2）栈顶，栈 1 的底在 V［1］，栈 2 的底在 V［m］，若两栈均采用顺序存储方式存储，则栈满的条件是（　）。

A.|top［2］-top［1］|=0

B.top［1］+1=top［2］

C.top［1］+top［2］=m

D.top［1］=top［2］

14.一个递归算法必须包括（　）。

A.递归部分

B.终止条件和递归部分

C.迭代部分

D.终止条件和迭代部分

15.执行完下列语句段后，i 值为（　）。intf（intx）{return（(x＆gt；0）？ x*f（x-1）：2)；} i=f（f（1））；

A.2　　B.4

C.8　　D.无限递归

16.表达式 a*（b+c）-d 的后缀表达式是（　）。

A.abcd*+-

B.abc+*d-

C.abc*+d-

D.-+*abcd

17.为了处理参数及返回地址，在递归过程或函数调用时，要用一种称为（　）的数据结构。

A.队列

B.多维数组

C.栈

D.线性表

18.若用一个大小为6的数组来实现循环队列，且当前rear和front的值分别为0和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

19.队尾已到达一维数组的最高下标，不能再插入元素，然而队中元素个数小于队列的长度，这种现象称作（　）。

A.上溢

B.下溢

C.假溢出

D.队列满

20.已知有一维数组A[0..m×n-1]，若要对应为m行、n列的矩阵，将元素A[k]（0≤k&lt；m×n）表示成矩阵的第i行、第j列的元素（0≤i&lt；m，0≤j&lt；n），则下面的对应关系是（　）。

A.i=k/n，j=k%m

B.i=k/m，j=k%m

C.i=k/n，j=k%n

D.i=k/m，j=k%n

综合题

1.简述栈、队列、循环队列的定义。

2.假设以I和O分别表示入栈和出栈操作，则对初态和终态均为空的栈操作可由I和O组成的序列表示。

（1）试指出判别给定序列是否合法的一般规则。

（2）两个不同合法序列（对同一输入序列）能否得到相同的输出元素序列？如能得到，请举例说明。

3.有5个元素，其入栈次序为A，B，C，D，E，在各种可能的出栈次序中，以元素C，D最先出栈（即C第一个且D第二个出栈）的次序有哪几个？

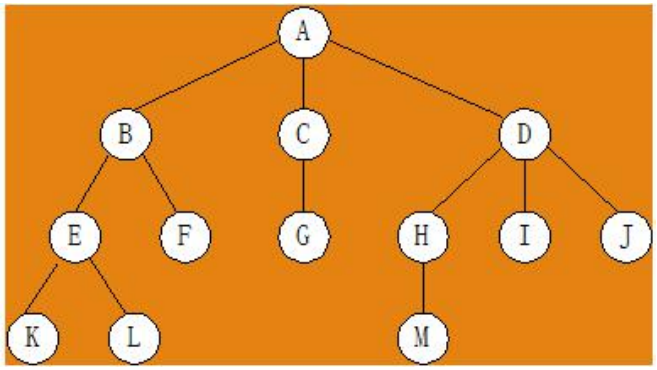
第三章　树与二叉树

树与二叉树 大纲考点

- 树的基本概念（☆☆）
- 二叉树定义以及主要性质（完全二叉树）（☆☆☆）
- 线索二叉树以及二叉排序树（☆☆）
- 赫夫曼树及其编码（☆☆）

基本内容

- 树的概念，相关概念。
- 二叉树的性质(5 条基本性质)与概念
- 特殊的二叉树
- 二叉树的存储
- 二叉树的遍历
- 线索二叉树
- 树的存储
- 树的遍历
- 哈夫曼（Huffman）树及其编码



一、基本概念

1.二叉树由(　)组成。

A.头指针、左子树、右子树

B.头结点、左子树、右子树

C.空指针、左子树、右子树

D.根结点、左子树、右子树

2.在完全二叉树中,若一个结点是叶结点,则它没有(　)。

A.左孩子结点

B.右孩子结点

C.左孩子结点、右孩子结点

D.左孩子结点、右孩子结点、右兄弟结点

3.线索二叉树是一种( )结构。

A.逻辑　　B.物理　　C.线性　　D.树形

二、二叉树的性质

题型分析:对于二叉树性质的掌握一定要深入,既要知道结论,也要知道结论的推导过程,例如二叉树中度为 0 的结点数等于度为 2 的结点数加 1。

4.已知一棵完全二叉树的第 6 层(设根为第 1 层)有 8 个叶子结点,则该完全二叉树的结点个数最多是(　)。

A.39　　B.52　　C.111　　D.119

5.在一棵树中,度为 3 的结点数为 2 个,度为 2 的结点数为 1 个,度为 1 的结点数为 2 个,则为 0 的结点数为 (　)个。

A.4　　B.5　　C.6　　D.7

6.深度为 5 的二叉树其结点数最多为(　)。

A.16　　B.30　　C.31　　D.32

7.n 个结点的线索二叉树上含有的线索数为(　)。

A.n-1　　B.n+1　　C.n　　D.2n

8.一棵二叉树高度为 h,所有结点的度或为 0,或为 2,则这棵二叉树最少有(　)个结点。

A.2h　　B.h+1　　C.2h+1　　D.1+2(h-1)

9.若一棵完全二叉树有 768 个结点,则该二叉树中叶结点个数是(　)。

A.257　　B.258　　C.384　　D.385

三、二叉树遍历

题型分析:本题型包含的考点比较多,既有简单的也有难的,大概有以下几种题目:
第一,对遍历过程的递归特性的深入理解。
第二,根据二叉树的遍历序列反推二叉树。
第三,遍历算法的相关应用。

10.对二叉树的结点从 1 开始进行连续编号,要求每个结点的编号大于其左、右孩子的编号,同一结点的左、右孩子中,其左孩子的编号小于其右孩子的编号,可采用(　)次序的遍历实现编号。

A.后序　　B.中序　　C.先序　　D.按层次遍历

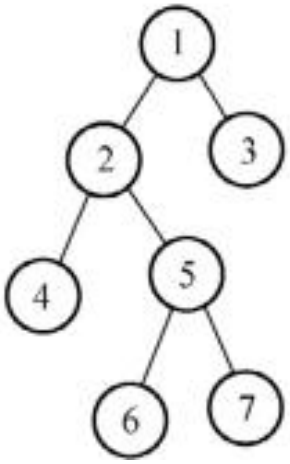
11.在二叉树的先序序列、中序序列和后序序列中,所有的叶子结点的先后顺序(　)。

A.都不相同
B.完全相同
C.先序和中序相同而与后序不同
D.中序和后序相同而与先序不同

12.某二叉树中序序列为 A,B,C,D,E,F,G,后序序列为 B,D,C,A,F,G,E,则该二叉树对应的森林包括(　)棵树。

A.1　　B.2　　C.3　　D.4

13.给定二叉树如图所示。设N代表二叉树的根,L代表根结点的左子树,R代表根结点的右子树。若遍历后的结点序列为3,1,7,5,6,2,4,则其遍历的方式是(　)。



A.LRN　　B.NRL　　C.RLN　　D.RNL

14.若一棵二叉树的先序遍历序列是a,e,b,d,c,后序遍历序列为b,c,d,e,a,则根结点的孩子结点(　)。

A.只有e　　B.有e,b　　C.有e,c　　D.无法确定

四、线索二叉树

题型分析:关于线索二叉树只需关注两方面的内容,其一是了解线索二叉树的概念及对二叉树进于线索化的目的,其二是知道如何在线索二叉树上查找结点的前驱和后继。

15.引入二叉线索树的目的是()。

A.加快查找结点的前驱或者后继的速度

B.为了能在二叉树中方便地进行插入和删除

C.为了方便地查找到双亲

D.使得二叉树的遍历结果唯一

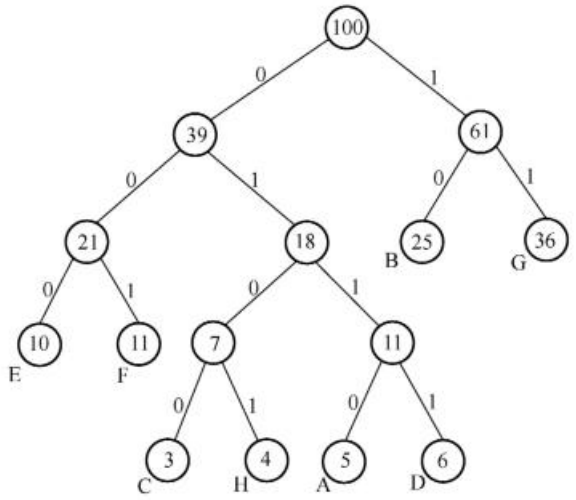
五、哈夫曼树的构造及其应用

题型分析:本题型主要考查学生对哈夫曼树的定义、特征、生成、编码知识点的掌握。

16.设给定权值总数有n个,其哈夫曼树的结点总数为()。

A.2n-1　　B.2n　　C.2n+1　　D.不能确定

17.已知某个字符串只可能出现八种字符(A~H),其中各个字符出现的次数分别为:A(5)、B(25)、C(3)、D(6)、E(10)、F(11)、G(36)、H(4)。现要对这个字符串用0、1两个数字进行前缀编码,要求使整个字符串编码后的总长度最短。请给出编码的方法和各个字符的编码(需要给出相应的哈夫曼树)。



六、二叉排序树

题型分析:二叉排序树必须要掌握定义、查找过程、查找效率、插入、删除、创建等内容

18.二叉查找树的查找效率与二叉树的（）有关,在（）时其查找效率最低。

A.结点的位置,结点太复杂

B.结点的多少,完全二叉树

C.树高度，结点太多

D.树形，呈单支

19.分别以下列序列构造二叉排序树,其中()序列构造出来的树与其他三个序列所构造的结果不同。

A.(100,80,90,60,120,110,130)

B.(100,120,110,130,80,60,90)

C.(100,60,80,90,120,110,130)

D.(100,80,60,90,120,130,110)

七、平衡二叉树

题型分析:平衡二叉树必须掌握两个方面内容:第一,单纯二叉树的定义;第二,当考查到插入.删除、创建等内容时,该平衡二叉树应该也是二叉排序树。当然也要清楚平衡二叉树的查找过程和查找效率

20.下列二叉排序树中查找效率最高的是()。

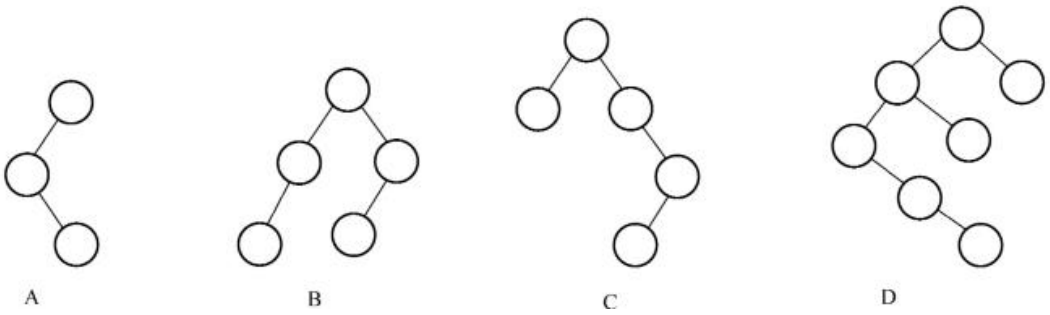
A.平衡二叉树

B.二叉查找树

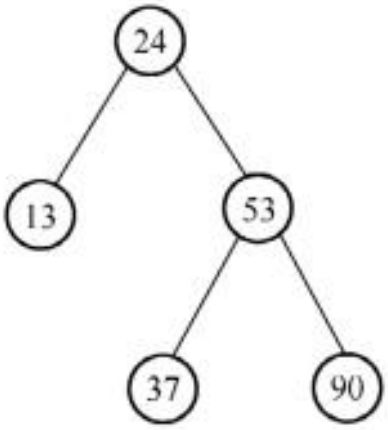
C.没有左子树的二叉排序树

D.没有右子树的二叉排序树

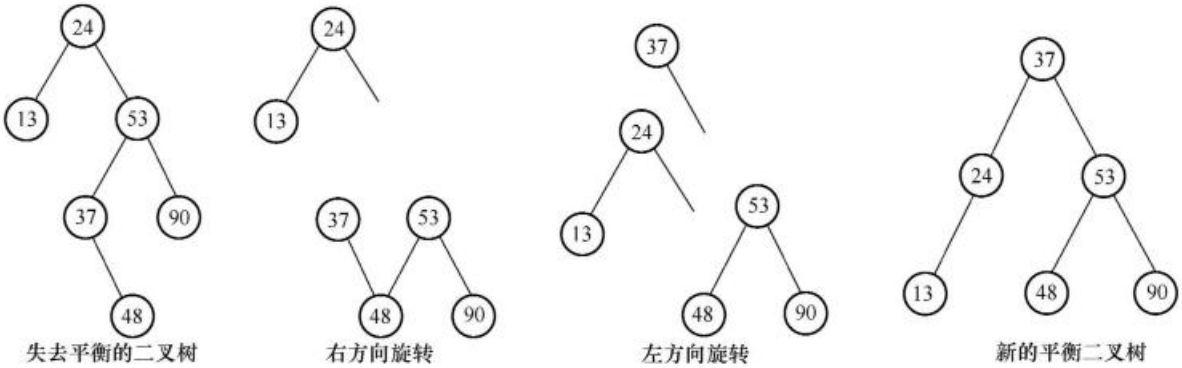
21.下图所示二叉排序树中满足平衡二叉树定义的是(　)。



21.下图所示的平衡二叉树中插入关键字 48 后得到一棵新平衡二叉树,在新平衡二叉树中关键字 37 所在结点的左、右子结点保存的关键字分别是(　)。



A.13,48　　B.24,48　　C.24,53　　D.24,90



八、树，二叉树，森林相互转化

题型分析:用孩子兄弟法将树向二叉树进行转化,这是一个基本的规则。关于森林和二叉树的转化只要遵守一条规则就行了:后面的树转化的二叉树作为它前面所对应的二叉树的右子树即可。

22.设森林 F 中有三棵树,第一、第二、第三棵树的结点个数分别为 M1,M2,和 M3。与森林 F

对应的二叉树根结点的右子树上的结点个数是(　)。

A.M1　　B.M3　　C.M1+M2　　D.M2+M3

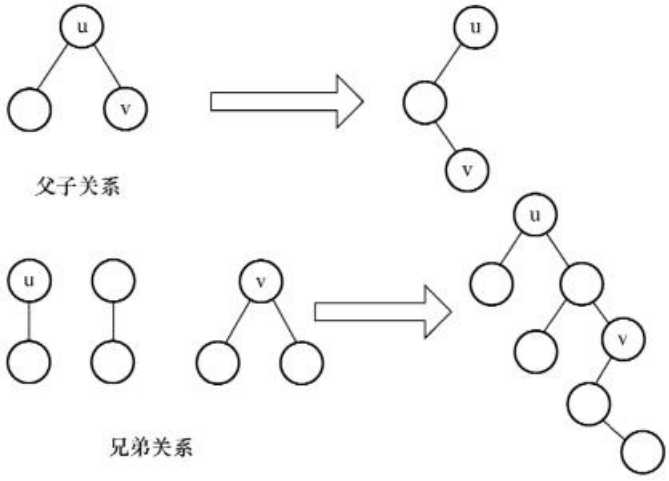
23.将森林转换为对应的二叉树,若在二叉树中,结点u是结点v的父结点的父结点,则在原来的森林中,u和v可能具有的关系是(　)。

1.父子关系

2.兄弟关系

3.u的父结点与v的父结点是兄弟关系

A.2　　B.1,2　　C.1,3　　D.1,2,3



真题

1.要使一棵非空二叉树的先序序列与中序序列相同，其所有非叶节点必须满足的条件是（　）.

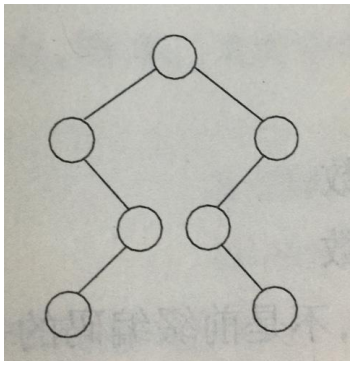
A.只有左子树

B.只有右子树

C.节点的度均为1

D.节点的度均为2

2.已知一棵二叉树的树形如下图所示，其后序序列为e，a，c，b，d，g，f，树中与节点a同层的节点是（　）.



A.c　　B.d　　C.f　　D.g

3.已知字符集{a，b，c，d，e，f，g，h}，若各字符的哈夫曼编码依次是0100，10，0000，0101，001，011，11，0001，则编码序列0100011001001011110101的译码结果是（　）.

A.a，c，g，a，b，f，h

B.a，d，b，a，g，b，b

C.a，f，b，e，a，g，d

D.a，f，e，e，f，g，d

4.若森林F有15条边，25个节点，则F包含树的个数是（　）.

A.8　　B.9　　C.10　　D.11

5.先序序列为a，b，c，d的不同二叉树的个数是（　）.

A.13　　B.14　　C.15　　D.16

6.下列选项给出的是从根分别到达两个叶节点路径上的权值序列,能属于同一棵哈夫曼树的是（　）.

A.24，10，5和24，10，7

C.2B.24，10，5和24，12，7

B.4，10，10和24，14，11

D.24，10，5和24，14，6

7.现有一棵无重复关键字的平衡二叉树（AVL树），对其进行中序遍历可得到一个降序序列。下列关于该平衡二叉树的叙述中，正确的是（　）.

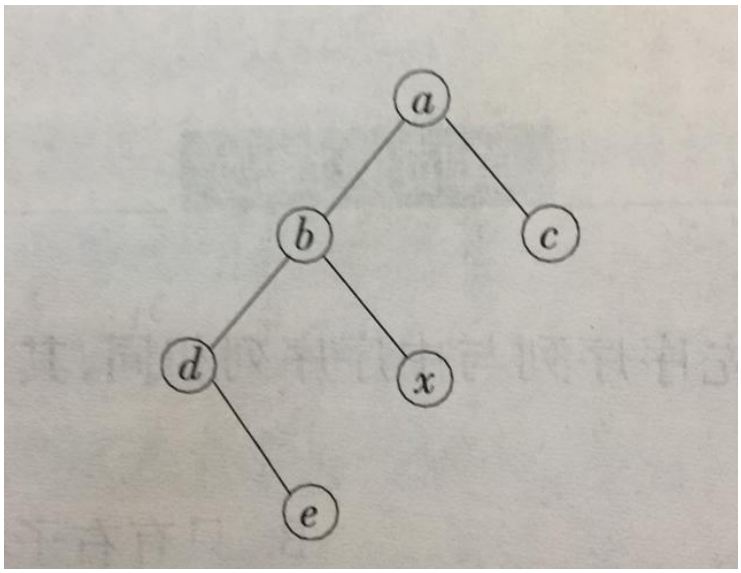
A.根节点的度一定为2

B.树中最小元素一定是叶节点

C.最后插入的元素一定是叶节点

D.树中最大的元素一定是无左子树

8.若对如下图所示的二叉树进行中序线索化,则节点x的左、右线索指向的节点分别是(　).



A.e，c　　B.e，a　　C.d，c　　D.b，a

9.5 个字符有如下 4 中编码方案，不是前缀编码的是（　）.

A.01，0000，0001，001，1

B.011，000，001，010，1

C.000，001，010，011，100

D.0，100，110，11110，1100

10.若 X 是后序线索二叉树中的叶节点,且 X 存在左兄弟节点 Y,则 X 的右线索指向的是(　).

A.X 的父节点

B.以 Y 为根的子树的最左下节点

C.X 的左兄弟节点 Y

D.以 Y 为根的子树的最右下节点

11.在任意一棵非空二叉排序树 T1 中，删除某节点 v 之后形成二叉排序树 T2，再将 v 插入形成二叉排序树 T3.下列关于 T1 和 T3 的叙述中，正确的是（　）.

Ⅰ.若 v 是 T1 的叶节点，则 T1 和 T3 不同

Ⅱ.若 v 是 T1 的叶节点，则 T1 和 T3 相同

Ⅲ.若 v 不是 T1 的叶节点，则 T1 和 T3 不同

Ⅳ.若 v 不是 T1 的叶节点，则 T1 和 T3 相同

A.仅Ⅰ、Ⅲ　　B.仅Ⅰ、Ⅳ

C.仅Ⅱ、Ⅲ　　D.仅Ⅱ、Ⅳ

12.若一棵完全二叉树有 768 个节点，则该二叉树中叶节点的个数是（　）

A.257　　B.258　　C.384　　D.385

13.若一棵二叉树的前序遍历序列和后序遍历序列分别为 1，2，3，4 和 4，3，2，1，则该二叉树的中序遍历序列不会是（　）.

A.1，2，3，4　　B.2，3，4，1

C.3，2，4，1　　D.4，3，2，1

14.对 n（n≥2）个权值均不相同的字符构造成哈夫曼树。下列关于该哈夫曼树的叙述中错误的是（）.

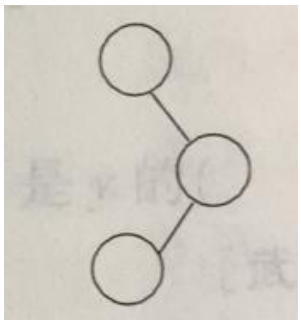
A.该树一定是一棵完全二叉树

B.树中一定没有度为 1 的节点

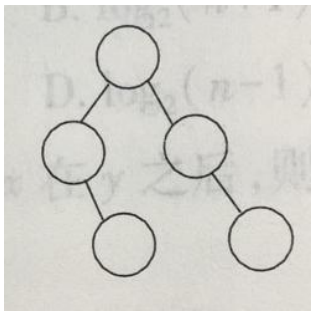
C.树中两个权值最小的节点一定是兄弟节点

D.树中任一非节点的权值一定不小于下一层任一节点的权值

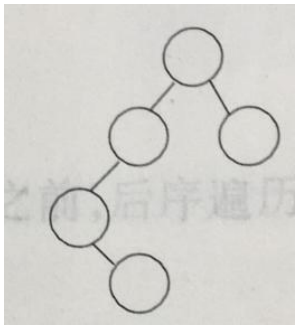
15. 下列二叉树中，满足平衡二叉树定义的是（　）.



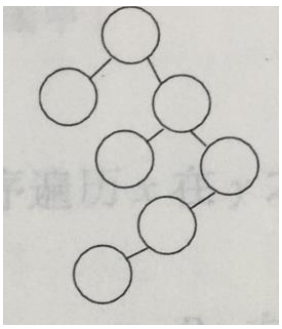
A



B



C



D

16. 若一棵二叉树具有 20 个度为 2 的节点，10 个度为 1 的节点，则度为 0 的节点的个数是（　）.

A. 10　　B. 11　　C. 21　　D. 30

17. 二叉树的第 i 层上最多有（　）个节点.

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

18. 有 n(n>0)个分支节点的满二叉树的深度是（　）.

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

19. 一棵二叉树的前序遍历序列为 A，B，C，D，E，F，G，它的中序遍历序列可能是（）.

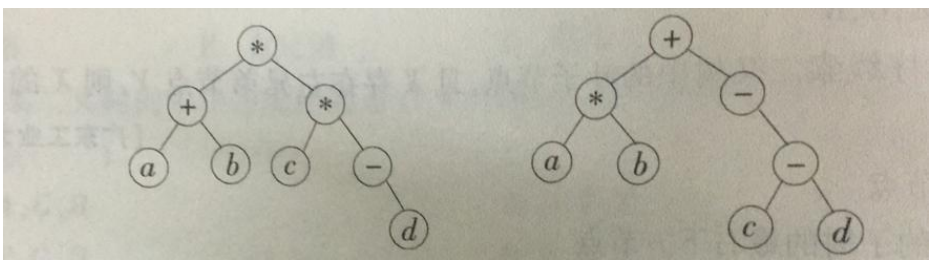
A. C，A，B，D，E，F，G

B. A，B，C，D，E，F，G

C. D，A，C，E，F，B，G

D. A，D，F，C，E，G，B

20. 请设计一个算法，将给定的表达式树（二叉树）转换为等价的中缀表达式（通过括号反应操作符的计算次序）并输出。例如，当下列两棵表达式树作为算法的输入时：



输出的等价中缀表达式分别为（a＋b）*（c*（－d））和（a*b）＋（－（c－d））。
二叉树节点定义如下：

```
typedefstructnode{
        chardata[10]；//存储操作数或操作符
        structnode*left，*right；
}BTree；
```

要求：

（1）给出算法的基本设计思想。

（2）根据设计思想，采用 C 或 C＋＋语言描述算法，关键之处给出注释。

21. 如果一棵非空 k（k≥2）叉树 T 中每个非叶节点都有 k 个孩子，则称 T 为正则 k 叉树。请回答下列问题，并给出推到过程。

（1）若 T 有 m 个非叶节点，则 T 中的叶节点有多少个？

（2）若 T 的高度为 h（单节点的树 h=1），则 T 的节点数最多为多少个？最少为多少个？

22. 二叉树的带权路径长度（WPL）是二叉树中所有叶节点的带权路径长度之和。给定一棵二叉树 T，采用二叉链表存储，节点结构为

其中，叶节点的 weight 域保存该节点的非负权值。设 root 为指向 T 的根节点的指针，请设计求 T 的 WPL 的算法。要求：

（1）给出算法的基本设计思想。

（2）根据 C 或 C++语言，给出二叉树节点的数据类型定义。

（3）根据设计思想，采用 C 或 C++语言描述算法，关键之处给出注释。

23. 若通信系统中只可能出现 5 种字符 A、B、C、D 和 E，其概率分别为 0.12、0.15、0.19、0.21 和 0.33。

（1）试设计哈夫曼编码。

（2）画出相应的哈夫曼树。

更多典型题

1. 在下面关于树的相关概念的叙述中，正确的是（　）。

A. 只有一个结点的二叉树的度为 1

B. 二叉树的度一定为 2

C. 二叉树的左右子树可任意交换

D. 深度为 K 的完全二叉树的结点个数小于或等于深度相同的满二叉树

2.已知一算术表达式的中缀形式为 A+B*C-D/E，后缀形式为 ABC*+DE/-，其前缀形式为（　）。

A.-A+B*C/DE　　B.-A+B*CD/E

C.-+*ABC/DE　　D.-+A*BC/DE

3.算术表达式 a+b*（c+d/e）转为后缀表达式后为（　）。

A.ab+cde/*　　B.abcde/+*+

C.abcde/*++　　D.abcde*/++

4.某二叉树的先序遍历序列为 IJKLMNO，中序遍历序列为 JLKINMO，则后序遍历序列是（　）。

A.JLKMNOI　　B.LKNJOMI

C.LKJNOMI　　D.LKNOJMI

5.设森林 F 对应的二叉树为 B，它有 m 个结点，B 的根为 P，P 的右子树结点个数为 n，森林 F 中第一棵树的结点个数是（　）。

A.m-n　　B.m-n-1

C.n+1　　D.条件不足，无法确定

6.二叉树若用顺序方法存储，则下列四种算法中运算时间复杂度最小的是（　）。

A.先序遍历二叉树

B.判断两个指定位置的结点是否在同一层上

C.层次遍历二叉树

D.根据结点的值查找其存储位置

7.设某二叉树中只有度为 0 和度为 2 的结点，如果此二叉树的高度为 100，那么此二叉树中所包含的结点数最少为（　）。

A.188　　B.200　　C.199　　D.201

8.树是结点的有限集合，一棵树中有（　）根结点。

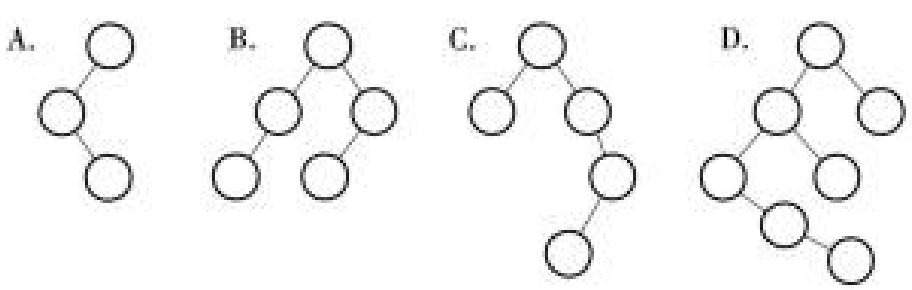
A.有 0 个或 1 个

B.有 0 个或多个

C.有且只有一个

D.有 1 个或 1 个以上

9.下列二叉排序树中，满足平衡二叉树定义的是（　）。



10.把树的根结点的层数定义为1，其他结点的层数等于其父结点所在层数加上1。设T是一棵二叉树，Ki和Kj是T中子结点数小于2的结点中的任意两个，它们所在的层数分别为λKi和λKj，当关系式|λKi-λKj|≤1一定成立时，则称T为一棵（　）。

A.满二叉树

B.二叉查找树

C.平衡二叉树

D.完全二叉树

11.设森林F中有三棵树，第一、第二、第三棵树的结点个数分别为M1、M2和M3。与森林F对应的二叉树根结点的右子树上的结点个数是（　）。

A.M1　　B.M1+M2　　C.M3　　D.M2+M3

12.若一棵二叉树具有10个度为2的结点，5个度为1的结点，则度为0的结点个数是（　）。

A.10　　B.11　　C.16　　D.不确定

13.具有10个叶结点的二叉树中有（　）个度为2的结点。

A.8　　B.9　　C.10　　D.11

14.在一棵度为3的树中，度为3的结点数为2个，度为2的结点数为1个，度为1的结点数为2个，则度为0的结点数为（　）个。

A.4　　B.5　　C.6　　D.7

15.已知一棵二叉树，共有n个结点，那么此二叉树的高度为（　）。

A.$n\log_2 n$　　B.$\log_2 n$

C.[log2n]+1　　D.不确定

16.已知一棵二叉树，第m层上最多含有结点数为（　）。

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

17.有关二叉树下列说法正确的是（　）。

A.二叉树就是度为2的树

B.一棵二叉树的度可以小于2

C.二叉树中至少有一个结点的度为2

D.二叉树中任何一个结点的度都为2

18.一棵二叉树的前序遍历序列为ABCDEFG，它的中序遍历序列可能是（　）。

A.CABDEFG　　B.ABCDEFG

C.DACEFBG　　D.BAECFDG

19.已知一个二叉树有1025个结点，那么由此推断二叉树的高h为（　）。

A.11　　B.10　　C.11～1025　　D.10～1024

20.一棵完全二叉树，共有n个结点，那么，其叶结点数共有（　）个。

A.n/2　　B.n　　C.（n-1）/2　　D.（n+1）/2

综合题

1.已知一个二叉树，用二叉链表形式存储，给出此二叉树建立过程算法（可不描述结构体）。

2.判别给定的二叉树是否是完全二叉树，并给出设计的算法（可不描述结构体）。

3.以孩子-兄弟表示法存储的森林的叶子结点数（要求描述结构）。

4.已知一棵二叉树的前序序列为：A，B，D，G，J，E，H，C，F，I，K，L；中序序列为：D，J，G，B，E，H，A，C，K，I，L，F。

（1）写出该二叉树的后序序列。

（2）画出该二叉树。

（3）求该二叉树的高度以及该二叉树中度为 2、1、0 的结点个数。

5.有 n 个结点的二叉树，已知叶结点个数为 n0。

（1）写出求度为 1 的结点的个数的 n1 的计算公式。

（2）若此树是深度为 k 的完全二叉树，写出 n 为最小的公式。

（3）若二叉树中仅有度为 0 和度为 2 的结点，写出求该二叉树结点个数 n 的公式。

6.已知一棵树的结点表示如下，其中各兄弟结点是依次出现的，画出对应的二叉树。

结点下标 A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
结点标记	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
父结点下标		0	0	0	1	1	2	2	3	3	4	5	5	6	7

7.在一棵表示有序集 S 的二叉搜索树（binarysearchtree）中，任意一条从根到叶结点的路径将 S 分为 3 部分：在该路径左边结点中的元素组成的集合 S1；在该路径上的结点中的元素组成的集合 S2；在该路径右边结点中的元素组成的集合 S3。S=S1∪S2∪S3。若对于任意的 a∈S1，b∈S2，c∈S3，是否总有 a≤b≤c？为什么？

8.假定用两个一维数组 L [N] 和 R [N] 作为有 N 个结点 1，2，…，N 的二叉树的存储结构。L [i] 和 R [i] 分别指示结点 i 的左儿子和右儿子；L [i] =0（R [i] =0）表示 i 的左（右）儿子为空。试写一个算法，由 L 和 R 建立一个一维数组 T [n]，使 T [i] 存放结点 i 的父亲；然后再写一个判别结点 U 是否为结点 V 的后代的算法。

9.试找出分别满足下面条件的所有二叉树：

（1）前序序列和中序序列相同。

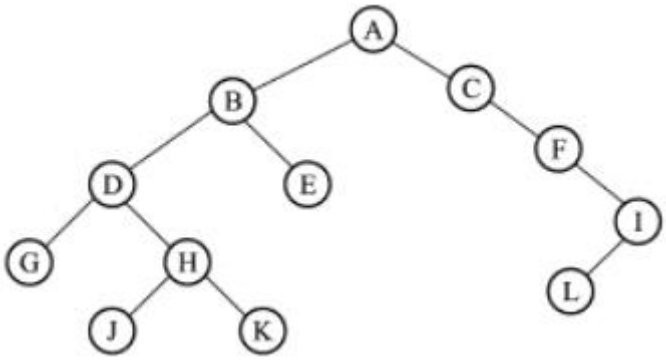
（2）中序序列和后序序列相同。

（3）前序序列和后序序列相同。

（4）前序、中序、后序序列均相同。

10.假设一个仅包含二元运算符的算术表达式以链表形式存储在二叉树 BT 中，写出计算该算术表达式值的算法。

11.画出如下图所示的二叉树所对应的森林。



12.下述编码中，哪一组不是前缀码？{00，01，10，11}，{0，1，00，11}，{0，10，110，111}

13.假设用于通信的电文由字符集{a，b，c，d，e，f，g，h}中的字母构成，这 8 个字母在电文中出现的概率分别为{0.07，0.19，0.02，0.06，0.32，0.03，0.21，0.10}。
（1）为这 8 个字母设计哈夫曼编码。
（2）若用三位二进制数（0～7）对这 8 个字母进行等长编码，则哈夫曼编码的平均码长是等长编码的百分之几？它使电文总长平均压缩多少？

14.有 n 个结点的完全二叉树存放在一维数组 A［1..n］中，试据此建立一棵用二叉链表表示的二叉树，根由 tree 指向。（可不定义结构体）

15.已知深度为 h 的二叉树采用顺序存储结构已存放于数组 BT［1..2h-1］中，请写一非递归算法，产生该二叉树的二叉链表结构。设二叉链表中链结点的构造为（lchild，data，rchild），根结点所在链结点的指针由 T 给出。

第四章 图

图大纲

- 大纲要求
- 图的基本概念（☆☆☆）
- 图的存储结构及基本操作（☆☆）
- 图的遍历（☆☆）
- 图的基本应用（☆☆☆）
-

基础知识

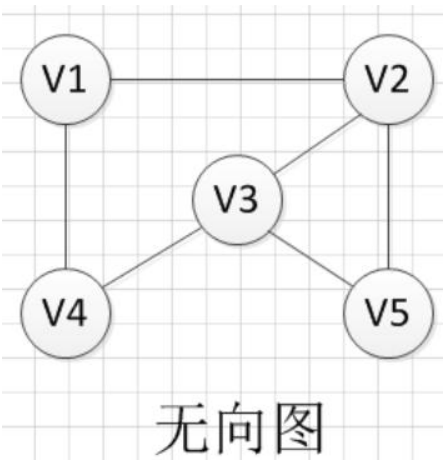
- 图的基本概念
- 图的存储及基本操作
- 图的遍历
- 图的基本应用

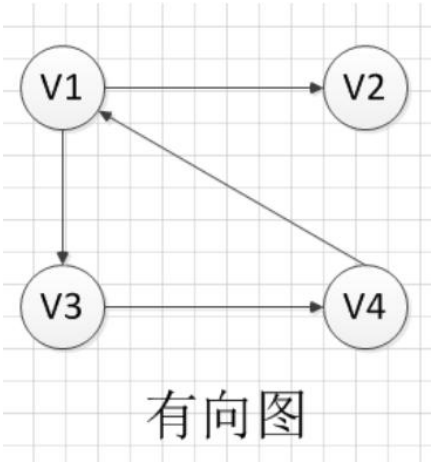
  最小生成树

  最短路径

  拓扑排序

  关键路径





一、基本概念

1.具有 n 个顶点的有向完全图有(　)条边。

A.n(n-1)/2

B.n(n-1)

C.n(n+1)/2

D.n(n+1)

2.若无向图 G-(V,E)中含有 7 个顶点,要保证图 G 在任何情况下都是连通的,则需要的边数最少是(　)。

A.6　　B.15

C.16　　D.21

3.关键路径是 AOE 网中(　)。

A.从源点到终点的最短路径

B.从源点到终点的最长路径

C.从源点到终点的边数最多的路径

D.从源点到终点边数最少的路径

4.下列关于 AOE 网的叙述中,不正确的是(　)。

A.关键活动不按期完成就会影响整个工程的完成时间

B.任何一个关键活动提前完成,那么整个工程将会提前完成

C.所有的关键活动提前完成,那么整个工程将会提前完成

D.某些关键活动若提前完成,那么整个工程将会提前完成

二、图的存储结构

题型分析:邻接矩阵、邻接表是图的两种很重要的存储结构,大家要了解其基本概念、存储细节、存储状态等。

5.下列图中的邻接矩阵属于对称矩阵的是(　)。

A.AOE 网　　B.有向图

C.无向图　　D.AOV 网

6.用邻接表存储图所用的空间大小(　)

A.与图的顶点数和边数有关

B.只与图的边数有关

C.只与图的顶点数有关

D.与边数的平方有关

7.已知有 6 个顶点(顶点编号为 0~5)的有向带权图 G,其邻接矩阵 A 为上三角矩阵,按行主序(行优先)保存在如下的一维数组中。

4	6	∞	∞	∞	1	∞	∞	∞	4	3	∞	∞	3	3

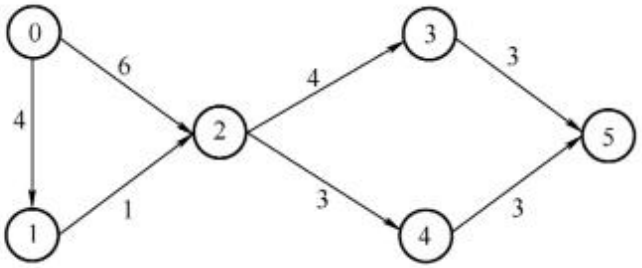
要求:

(1)写出图 G 的邻接矩阵 A.

(2)画出有向带权图 G。

(3)求图 G 的关键路径,并计算该关键路径的长度。

$$\begin{bmatrix} 0 & 4 & 6 & \infty & \infty & \infty \\ \infty & 0 & 1 & \infty & \infty & \infty \\ \infty & \infty & 0 & 4 & 3 & \infty \\ \infty & \infty & \infty & 0 & \infty & 3 \\ \infty & \infty & \infty & \infty & 0 & 3 \\ \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$



8.在有向图的十字链表表示法中弧结点的个数是图中弧的个数的()倍。

A.1　　B.2　　C.1/2　　D.不确定

三、图的遍历

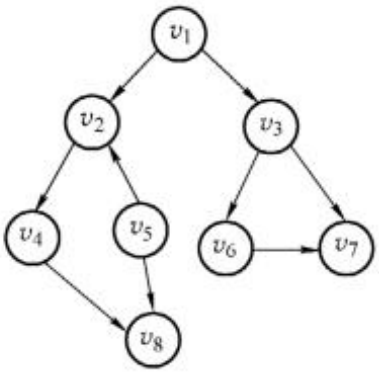
题型分析:这类题型大家要注意几个细节问题:

第一,如果出题人给你一个图形,然后让你遍历即使告诉你一个起始顶点,也不能得到一个唯一的遍历序列。

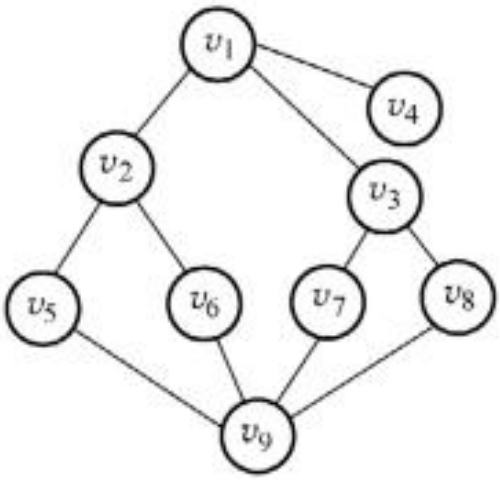
第二,如果出题人给的是一个图的存储结构,然后告诉你访问的起始顶点,这时遍历序列就唯一了。

第三,深度优先遍历是借助于栈这个数据结构的,而广度优先遍历是借助于队列这个数据结构的。

9.下图所示有向图的深度、广度优先遍历的结果是(　)。



10.下图所示的无向图的深度、广度优先遍历的结果是().



11.对有 n 个顶点、e 条边且使用邻接表存储的有向图进行广度优先遍历,其算法时间复杂度是(　)。

A.O(n)

B.O(e)

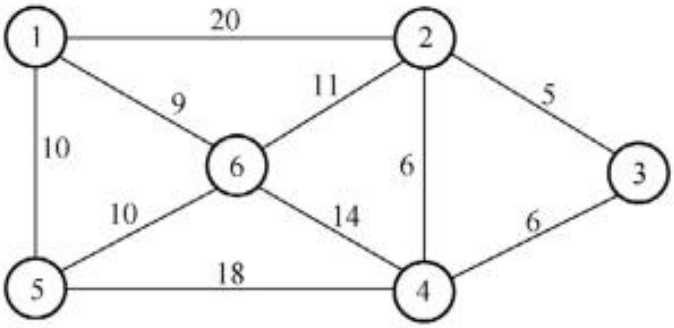
C.O(n+e)

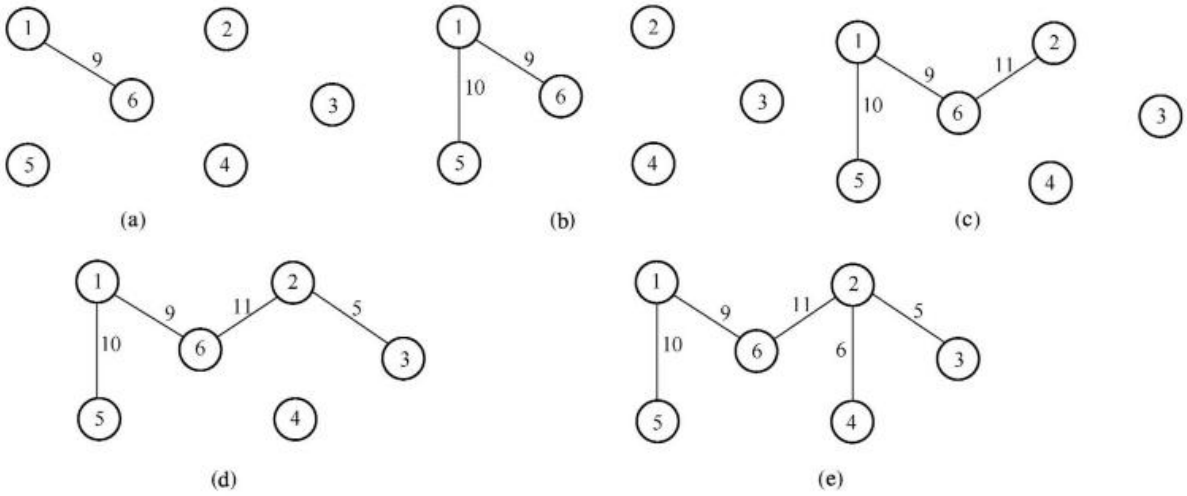
D.O(n*e)

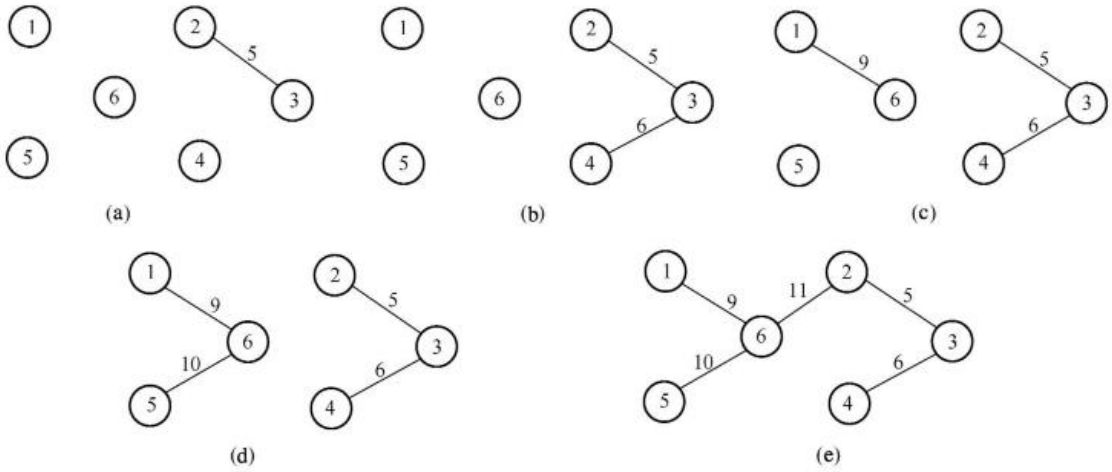
四、最小生成树

题型分析:对于最小生成树的构造,大家只需熟练掌握两种算法即可,即普里姆算法和克鲁斯卡·尔算法。

12.已知一个无向图,要求分别用普里姆算法和克鲁斯卡尔算法生成最小生成树(假设以 1 为起点,并画出构造过程)。



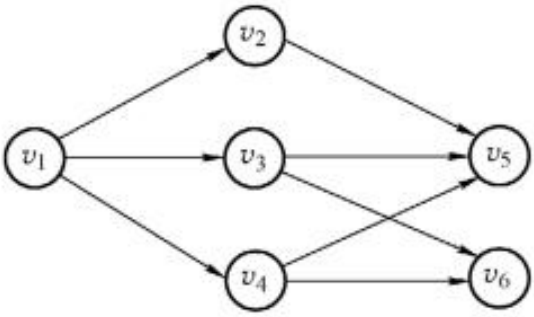




五、拓扑排序

题型分析:关于拓扑排序,考生首先要明白其目的,就是检查工作进度安排是否合理。具体来说就是如果对于一个图来说,对其进行拓扑排序之后,如果图中所有结点都在拓扑排序序列中,则说明该图无环,也就是说,该图所对应的工作进度安排是合理的。

13.有向图如图所示



要求:(1)给出拓扑排序的算法代码;(2)按照你的算法给出图执行的结果。

```
void  topsort(hdnodes  graph[],int  n){
  int i,j,k,top;
  node_pointer  ptr;
  top=-1;
  for(i=0; i<n; i++){
    if(! graph[i].count) {
        graph [i] .count=top;
        top=i;
    }
}
for( i=0;i<n;i++){
    if(top==-1) {
        printf("有一个环\n");
        return;
    }
    else {
        j=top;
        top=graph [j] .count;
        printf("V%d,", j);
        for(ptr=graph.link;ptr;ptr=ptr->link){
            k=ptr->vertex;
            graph[k].count-- ;
            if(graph[k].count==0){
                graph[k].count=top;
                top=k;
            }
          }
      }
  }
}
```

14.若用邻接矩阵存储有向图,矩阵中主对角线以下的元素均为零,则关于该图的拓扑排序的结论是()。

A.存在,且唯一

B.存在,且不唯一

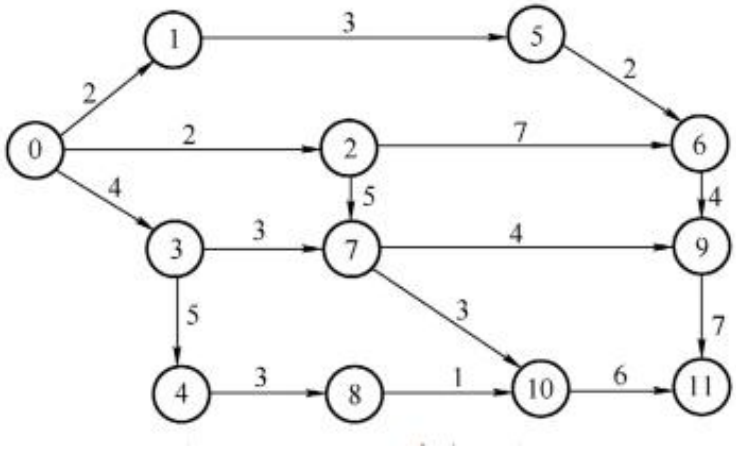
C.存在,可能不唯一

D.无法确定是否存在

六、关键路径

题型分析:求关键路径的目的就是寻找影响整个工程进度的关键性活动,然后对这些活动进行分析,看哪些活动的效率可以提高,从而达到缩短整个工期的目的。

15.某一工程作业的网络如图所示



其中箭头表示作业,箭头旁边的数字表示完成作业所需的天数,箭头前后的圆圈表示事件,圆圈中的数字表示事件的编号,用事件编号的序号表示进行作业的路径,问:

(1)完成此工程的关键路径是什么?

(2)完成该工程至少需要多少天?

(3)该工程中具有最大充裕天数的事件是什么事件?充裕天数是多少?

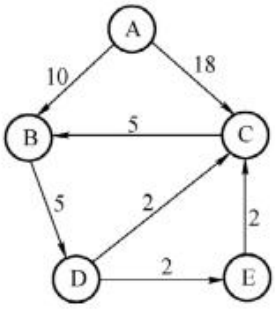
(4)关键路径上的事件的充裕天数是多少?

	0	1	2	3	4	5	6	7	8	9	10	11
$v_e(i)$	0	2	2	4	9	5	9	7	12	13	13	20
$v_l(i)$	0	4	2	4	9	7	9	9	12	13	14	20
v_l-v_e	0	2	0	0	0	2	0	2	0	0	1	0

七、最短路径

题型分析:关于最短路径,考生首先要掌握单源点到其余各顶点的最短路径的求法,迪杰斯特拉算法就是其中的一个方法。其次,要掌握图中所有顶点中任意两点之间的最短路径的求法。

16.以下图为例



按迪杰斯特拉算法计算从顶点 A 到其他各个顶点的最短路径和最短路径长度。

$$\begin{bmatrix} 0 & 10 & 18 & \infty & \infty \\ \infty & 0 & \infty & 5 & \infty \\ \infty & 5 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & 2 \\ \infty & \infty & 2 & \infty & 0 \end{bmatrix}$$

第一步：

	A	B	C	D	E
s	1	0	0	0	0
dist	0	10	18	∞	∞

该路径为：A→A；A→B；A→C；A无法到D和E。

第二步：

	A	B	C	D	E
s	1	1	0	0	0
dist	0	10	18	15	∞

该路径为：A→A；A→B；A→C；A→B→D；A无法到E。

第三步：

	A	B	C	D	E
s	1	1	0	1	0
dist	0	10	17	15	17

该路径为：A→A；A→B；A→B→D→C；A→B→D；A→B→D→E。

第四步：

	A	B	C	D	E
s	1	1	1	1	1
dist	0	10	17	15	17

该路径为：A→A；A→B；A→B→D→C；A→B→D；A→B→D→E。

真题

1.已知无向图 G 含有 16 条边，其中度为 4 的顶点个数为 3，度为 3 的顶点个数为 4，其他顶点的度均小于 3.图 G 所含的顶点个数至少是（　）.

A.10　　B.11　　C.13　　D.15

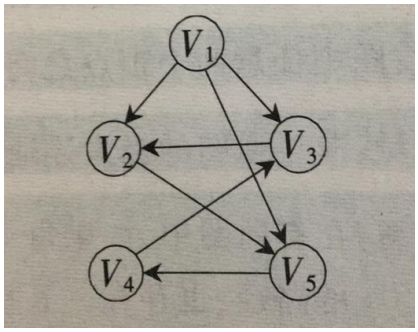
2.下列选项中,不是下图深度优化搜索序列的是().

A.V1，V5，V4，V3，V2

B.V1，V3，V2，V5，V4

C.V1，V2，V5，V4，V3

D.V1，V2，V3，V4，V5



3.若将 n 个 i 顶点、e 条弧的有向图采用邻接表存储，则拓扑排序算法的时间复杂度是（ ）.

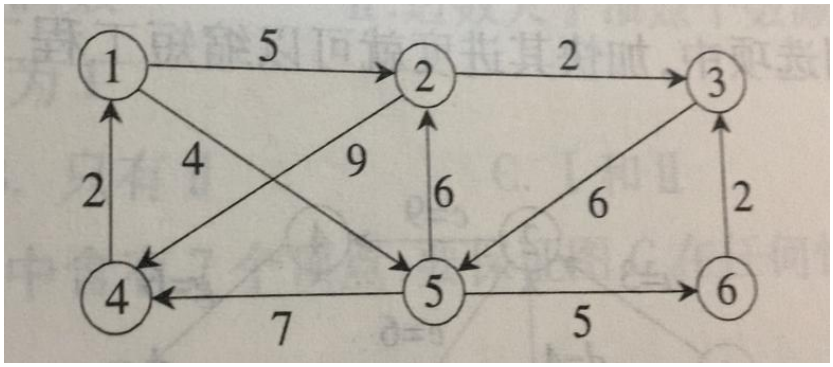
A.O(n)

B.O(n+e)

C.O(n2)

D.O(n*e)

4.使用迪杰斯特拉（Dijkstra）算法求下图中从顶点 1 到其他各顶点的最短路径，依次得到的各最短路径的目标定点是（ ）.



A.5，2，3，4，6

B.5，2，3，6，4

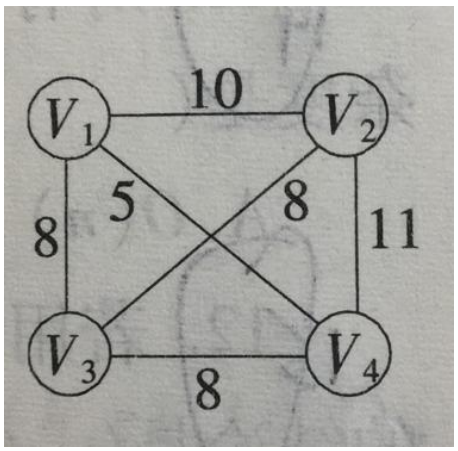
C.5，2，4，3，6

D.5，2，6，3，4

5.设有向图 G=(V，E)，顶点集 V={V0，V1，V2，V3}，边集 E={<V0，V1>，<V0，V2>，<V0，V3>，<V1，V3>}。若顶点 V0 开始对图进行深度优先遍历，则可能得到的不同遍历序列个数是（）.

A.2　　B.3　　C.4　　D.5

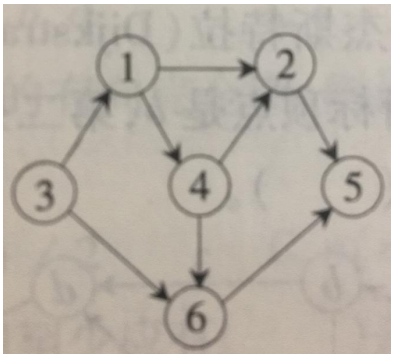
6.求如右图所示带权图的最小（代价）生成树时，可能是克鲁斯卡尔（Kruskal）算法第 2 次选中，但不是普里姆（Prim）算法（从 V4 开始）第二次选中的边是（）.



A.（V1，V3）　　B.（V1，V4）

C.（V2，V3）　　D.（V3，V4）

7.对如下图所示的有向图进行拓扑排序，得到的拓扑序列可能是（）.



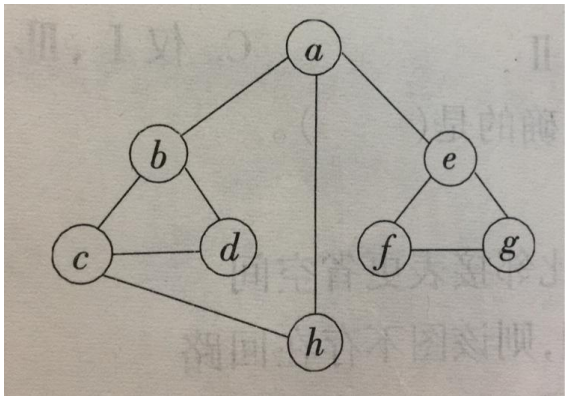
A.3，1，2，4，5，6
B.3，1，2，4，6，5
C.3，1，4，2，5，6
D.3，1，4，2，6，5

8.设图的邻接矩阵 A 如下图所示。各顶点的度依次是（）.

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

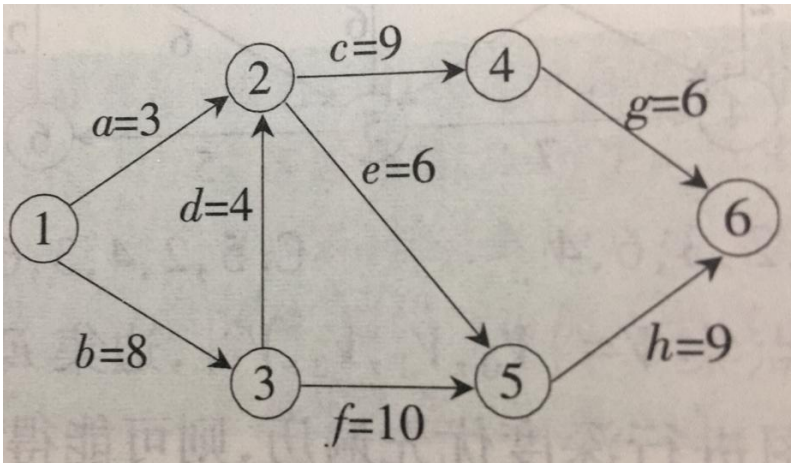
A.1，2，1，2　　B.2，2，1，1
C.3，4，2，3　　D.4，4，2，2

9.若对如下无向图进行遍历，则下列选项中，不是广度优先遍历序列的是（）.



A.h，c，a，b，d，e，g，f
B.e，a，f，g，b，h，c，d
C.d，b，c，a，h，e，f，g
D.a，b，c，d，h，e，f，g

10.下图所示 AOE 网表示一项包含 8 个活动的工程。通过同时加快若干活动的进度可以缩短整个工程的工期。下列选项中，加快其进度就可以缩短工程工期的是（）.



A.c 和 e　　B.d 和 e　　C.f 和 d　　D.f 和 h

11.对有 n 个节点、e 条边且使用邻接表存储的有向图进行广度优先遍历，其算法时间复杂度是（）.

A.O（n）　　B.O（e）

C.O（n+e）　　D.O（n*e）

12.若用邻接矩阵存储有向图，矩阵中主对角线以下的元素均为 0，则关于该图拓扑序列的结论是（）.

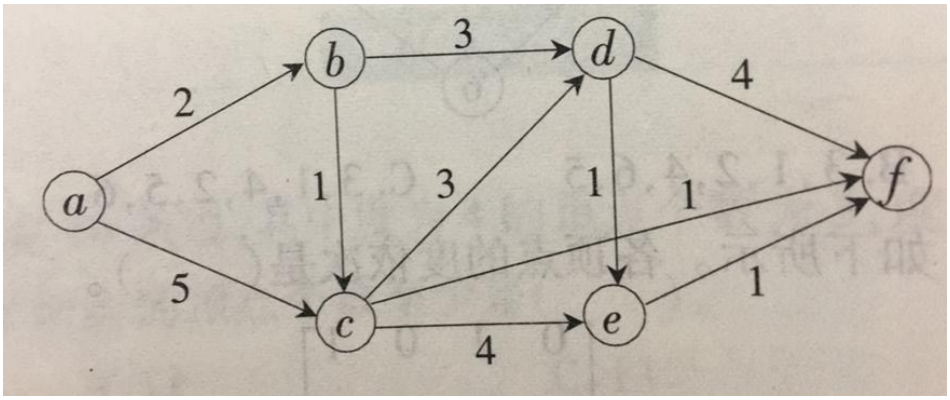
A.存在，且唯一

B.存在，且不唯一

C.存在，可能不唯一

D.无法确定是否存在

13.对如下有向带权图，若采用迪杰斯特拉（Dijkstra）算法求从源点 a 到其他各顶点的最短路径，则得到的第一条最短路径的目标顶点是 b，第二条最短路径的目标顶点是 c，后续得到的其余最短路径的目标顶点依次是（）.



A.d，e，f　　B.e，d，f

C.f，d，e　　D.f，e，d

14.下列关于最小生成树的叙述中，正确的是（）.

Ⅰ.最小生成树的代价唯一

Ⅱ.所有权值最小的边一定会出现在所有最小生成树中

Ⅲ.使用普里姆（Prim）算法从不同顶点开始得到的最小生成树一定相同

Ⅳ.使用普里姆算法和克鲁斯卡尔（Kruskal）算法得到的最小生成树总不相同

A.仅Ⅰ　　B.仅Ⅱ

C.仅Ⅰ、Ⅲ　　D.仅Ⅱ、Ⅲ

15.下列关于图的叙述中，正确的是（）.

Ⅰ.回路是简单路径

Ⅱ.存储稀疏图，用邻接矩阵比邻接表更省空间

Ⅲ.若有向图中存在拓扑序列，则该图不存在回路

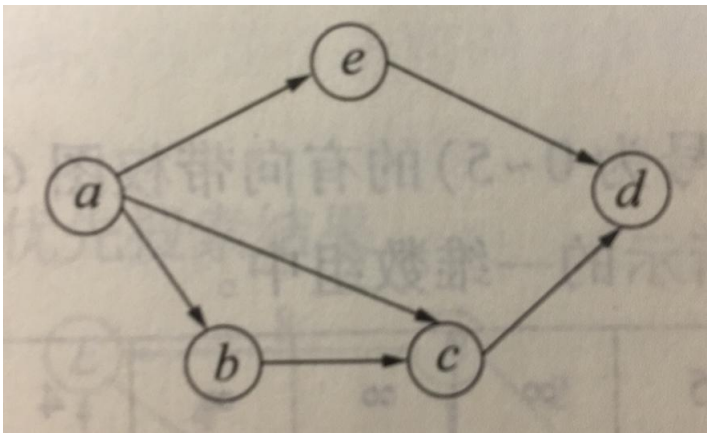
A.仅Ⅱ　　B.仅Ⅰ、Ⅱ

C.仅Ⅲ　　D.仅Ⅰ、Ⅲ

16.若无向图G=（V，E）中含有7个顶点，要保证图G在任何情况下都是连通的，则需要的边数最少是（）.

A.6　　B.15　　C.16　　D.21

17.对下图进行拓扑排序，可以得到不同的拓扑序列的个数是（）.



A.4　　B.3　　C.2　　D.1

18.无向图G=（V，E），其中V={a，b，c，d，e，f}，E={（a，b），（a，e），（a，c），（b，e），（c，f），（f，d），（e，d）}，对该图进行深度优先遍历，得到的顶点序列正确的是（）.

A.a，b，e，c，d，f

B.a，c，f，e，b，d

C.a，e，b，c，f，d

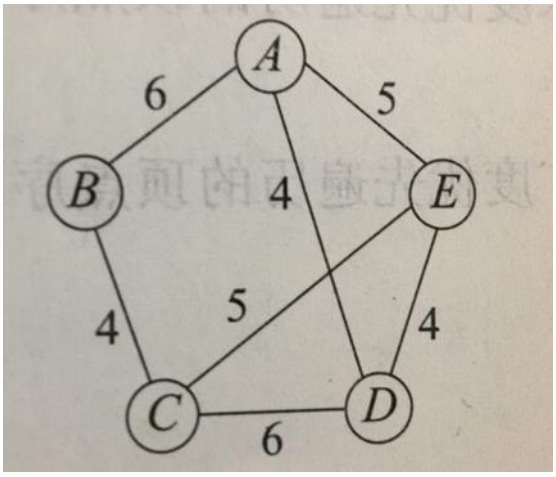
D.a，e，d，f，c，b

19.使用Prim（普里姆）算法求带权连通图的最小（代价）生成树（MST）。请回答下列问题：

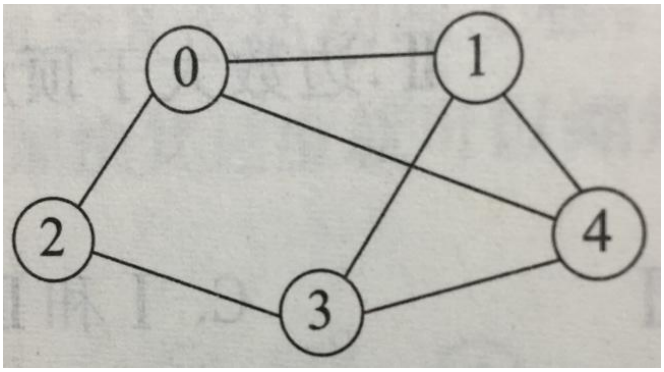
（1）对下列图G，从顶点A开始求G的MST，依次给出按算法选出的边。

（2）图 G 法人 MST 是唯一的吗？

（3）对任意的带权连通图，满足什么条件时，其 MST 是唯一的？



20. 已知含有 5 个顶点的图 G 如图所示



请回答下列问题：

（1）写出图 G 的邻接矩阵 A（行、列下标从 0 开始）

（2）求 A 的平方，矩阵 A2 中位于 0 行 3 列元素值的含义是什么？

（3）若已知具有 n（n≥2）个顶点的图的邻接矩阵为 B，则 B 的 m 次方（2≤m≤n）中非零元素的含义是什么？

21. 已知有 6 个顶点（顶点的编号为 0~5）的有向带权图 G，其邻接矩阵 A 为上三角矩阵，按行为主序（行优先）保存在如下图所示的一维数组中。

4	6	∞	∞	∞	5	∞	∞	∞	4	3	∞	∞	3	3

要求：

（1）写出图 G 的邻接矩阵 A。

（2）画出有向带权图 G。

（3）求图 G 的关键路径，并计算该关键路径的长度。

更多典型题

1.下面是一个求最小生成树的算法，其中 G 是连通无向图，T 是所求的生成树。试问该算法是哪一种求最小生成树的算法？（　）

A.Prim（普里姆）算法

B.Kruskal（克鲁斯卡尔算法）

C.罗巴赫算法

D.其他算法

2.邻接表是图的一种（　）。

A.顺序存储结构

B.链接存储结构

C.索引存储结构

D.散列存储结构

3.下面试图对图中路径进行定义，说法正确的是（　）。

A.由顶点和相邻顶点序列构成的边所形成的序列

B.由不同顶点所形成的序列

C.由不同边所形成的序列

D.上述定义都不是

4.无向图中顶点个数为 n，那么边数最多为（　）。

A.n-1　　B.n（n-1）/2

C.n（n+1）/2　　D.n2

5.在一个具有 n（n&gt；0）个顶点的连通无向图中，至少需要的边数是（　）。

A.n　　B.n+1

C.n-1　　D.n/2

6.以下叙述中正确的是（　）。

Ⅰ.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

Ⅱ.连通图的广度优先搜索中一般要采用队列来暂存访问过的顶点

Ⅲ.图的深度优先搜索中一般要采用栈来暂存访问过的顶点

A.Ⅰ，Ⅱ　　B.Ⅱ，Ⅲ

C.Ⅰ，Ⅲ　　D.Ⅰ，Ⅱ，Ⅲ

7.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∞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

8.在一个无向图中，所有顶点的度之和等于边数的（　）倍。

A.1/2　　B.1　　C.2　　D.4

9.采用邻接表存储的图的深度优先遍历算法类似于二叉树的（　）算法。

A.前序遍历　　B.中序遍历　　C.后序遍历　　D.按层遍历

10.任何一个无向连通图（　）最小生成树。

A.只有一棵　　B.有一棵或多棵　　C.一定有多棵　　D.可能不存在

11.判断有向图是否存在回路，除了可以利用拓扑排序方法外，还可以利用的是（　）。

A.求关键路径的方法　　B.求最短路径的迪杰斯特拉方法

C.深度优先遍历算法　　D.广度优先遍历算法

12.有 n 个顶点 e 条边的无向图，采用邻接表存储时，有（　）个表头结点，有（　）个链表结点。

A.n，2e　　B.n，2e+1　　C.n-1，2e　　D.n-1，2e+1

13.对于由 n 个顶点组成的有向完全图来说，图中共包含（　）条边，对于由 n 个顶点组成的无向完全图来说，图中共包含（　）条边。

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

14.在一个（　）图中寻找拓扑序列的过程称为（　）。

A.有向，拓扑排序　　B.无向，拓扑排序

C.有向，最短路径搜索　　D.无向，最短路径搜索

15.用邻接矩阵 A 表示图，判定任意两个顶点 vi 和 vj 之间是否有长度为 m 的路径相连，则只要检查（　）的第 i 行第 j 列的元素是否为零即可。

A.mA

B.A

C.A^m

D.Am-1

16.当各边上的权值（　）时，BFS 算法可用来解决单源最短路径问题。

A.均相等　　B.均互不相等　　C.不一定相等　　D.不确定

17.下面关于图的存储结构的叙述中正确的是（　）。

A.用邻接矩阵存储图占用空间大小只与图中顶点数有关，与边数无关

B.用邻接矩阵存储图占用空间大小只与图中边数有关，与顶点数无关

C.用邻接表存储图占用空间大小只与图中顶点数有关，与边数无关

D.用邻接表存储图占用空间大小只与图中边数有关，与顶点数无关

18.在有向图 G 的拓扑序列中，若顶点 vi 在顶点 vj 之前，则下列情形不可能出现的是（　）。

A.G 中有弧＆lt；vi，vj＆gt；　　B.G 中有一条从 vi 到 vj 的路径

C.G 中没有弧＆lt；vi，vj＆gt；　　D.G 中有一条从 vj 到 vi 的路径

19.以下关于图的说法中正确的是（　）。

Ⅰ.一个有向图的邻接表和逆邻接表中的结点个数一定相等

Ⅱ.用邻接矩阵存储图，所占用的存储空间大小只与图中结点个数有关，而与图的边数无关

Ⅲ.无向图的邻接矩阵一定是对称的，有向图的邻接矩阵一定是不对称的

A.Ⅰ，Ⅱ　　B.Ⅱ，Ⅲ　　C.Ⅰ，Ⅲ　　D.仅有Ⅱ

20.下列关于 AOE 网的叙述中，不正确的是（　）。

A.关键活动不按期完成就会影响整个工程的完成时间

B.任何一个关键活动提前完成，那么整个工程将会提前完成

C.所有的关键活动提前完成，那么整个工程将会提前完成

D.某些关键活动提前完成，那么整个工程将会提前完成

21.一个二部图的邻接矩阵 A 是一个（　）类型的矩阵。

A.n×n 矩阵　　B.分块对称矩阵　　C.上三角矩阵　　D.下三角矩阵

22.求解最短路径的 Floyd 算法的时间复杂度为（　）。

A.O（n）　　B.O（n+c）　　C.O（n2）　　D.O（n3）

综合题

1.证明：具有 n 个顶点和多于 n-1 条边的无向连通图 G 一定不是树。

2.证明：对有向图的顶点适当地编号，可使其邻接矩阵为下三角形且主对角线为全 0 的充要条件是该图为无环图。

3.关于图（Graph）的一些问题：

（1）有 n 个顶点的有向强连通图最多有多少条边？最少有多少条边？

（2）表示有 1000 个顶点、1000 条边的有向图的邻接矩阵有多少个矩阵元素？是否为稀疏矩阵？

4.如何对有向图中的顶点号重新安排可使得该图的邻接矩阵中所有的 1 都集中到对角线以上？

5.假定图 G=（V，E）是有向图，V=｛1，2，…，N｝，N≥1，G 以邻接矩阵方式存储，G 的邻接矩阵为 A，即 A 是一个二维数组。如果 i 到 j 有边，则 A［i，j］=1，否则 A［i，j］=0。请给出一个算法思想，该算法能判断 G 是否是非循环图（即 G 中是否存在回路），要求算法的时间复杂性为 O（n2）。

6.对一个图进行遍历可以得到不同的遍历序列，那么导致得到的遍历序列不唯一的因素有哪

些？

7.对于一个使用邻接表存储的有向图 G，可以利用深度优先遍历方法，对该图中结点进行拓扑排序。其基本思想是：在遍历过程中，每访问一个顶点，就将其邻接到的顶点的入度减 1，并对其未访问的、入度为 0 的邻接到的顶点进行递归。

（1）给出完成上述功能的图的邻接表定义。

（2）定义在算法中使用的全局辅助数组。

（3）写出在遍历图的同时进行拓扑排序的算法。

第五章　查找

查找 大纲要求

- 查找的基本概念（☆☆）
- 顺序，分块，折半等查找方法（☆☆☆）
- B,B+树的基本概念和操作（☆☆）
- 散列（Hash）表（☆☆☆）

基本概念

- 查找，顺序查找的基本内容
- 二叉排序树，基本操作
- 平衡二叉树
- B-树，基本性质
- B+树，与 B-树的区别
- 散列表，散列函数冲突处理

一、基本概念

1.对线性表进行折半查找,要求线性表必须( )。

A.以顺序方式存储

B.以链式方式存储

C.以顺序方式存储,且结点按关键字有序排序

D.以链式方式存储,且结点按关键字有序排序

2.顺序查找法适合于存储结构为( )的线性表。

A.散列存储

B.顺序存储或链式存储

C.压缩存储

D.索引存储

3.下列叙述中,不符合 m 阶 B 树定义求的是（)。

A.根结点最多有 m 棵子树在同一层上

B.所有叶子结点都在同一层上

C.各结点内关键字均升序或降序排列

D.叶子结点之间通过指针链接

4.将 10 个元素散列到 100000 个单元的哈希表中,则(　)产生冲突。

A.不可能　　B.一定会

C.有可能　　D.不知道

二、折半查找过程及效率

题型分析:考生需熟练掌握折半查找的过程:先找到中间元素,然后与中间元表比较,这时会有三种结果,根据结果确定如何进行下一步操作。同时,考生还应该对这个过程进行描述,就是用二叉判定树对折半查找过程进行描述。

5.在顺序表(8,11,15,19,25,26,30,33,42,48,50)中,用二分(折半)法查找关键码值 20,需做的关键码比较次数为()。

A.2　　B.3　　C.4　　D.5

6.已知一个长度为 16 的顺序表 L,其元素按关键字有序排列,若采用折半查找法查找一个不存在的元素,则比较次数最多是()。

A.4　　B.5　　C.6　　D.7

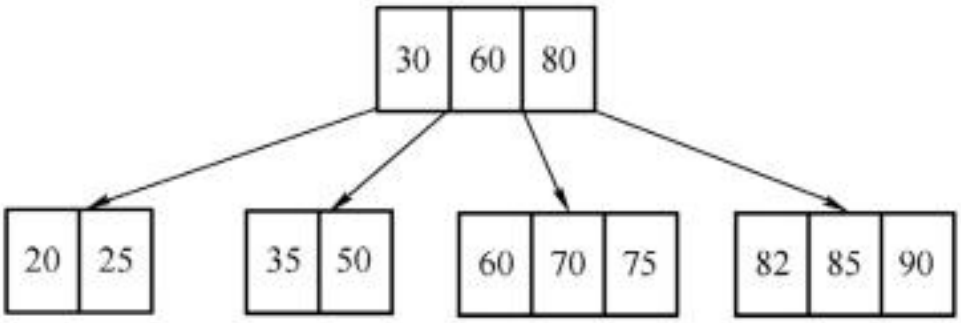
7.具有 12 个关键字的有序表,折半查找的平均查找长度为()。

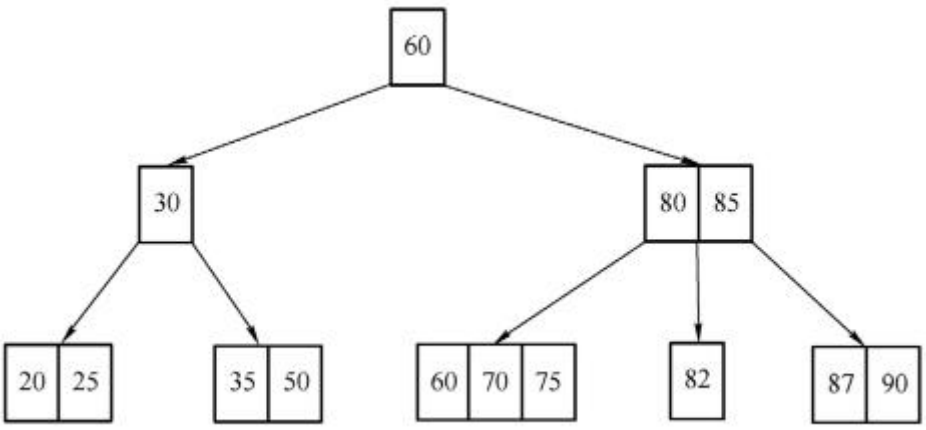
A.3.1　　B.4　　C.2.5　　D.5

三、B-(B)树节点的分裂和合并

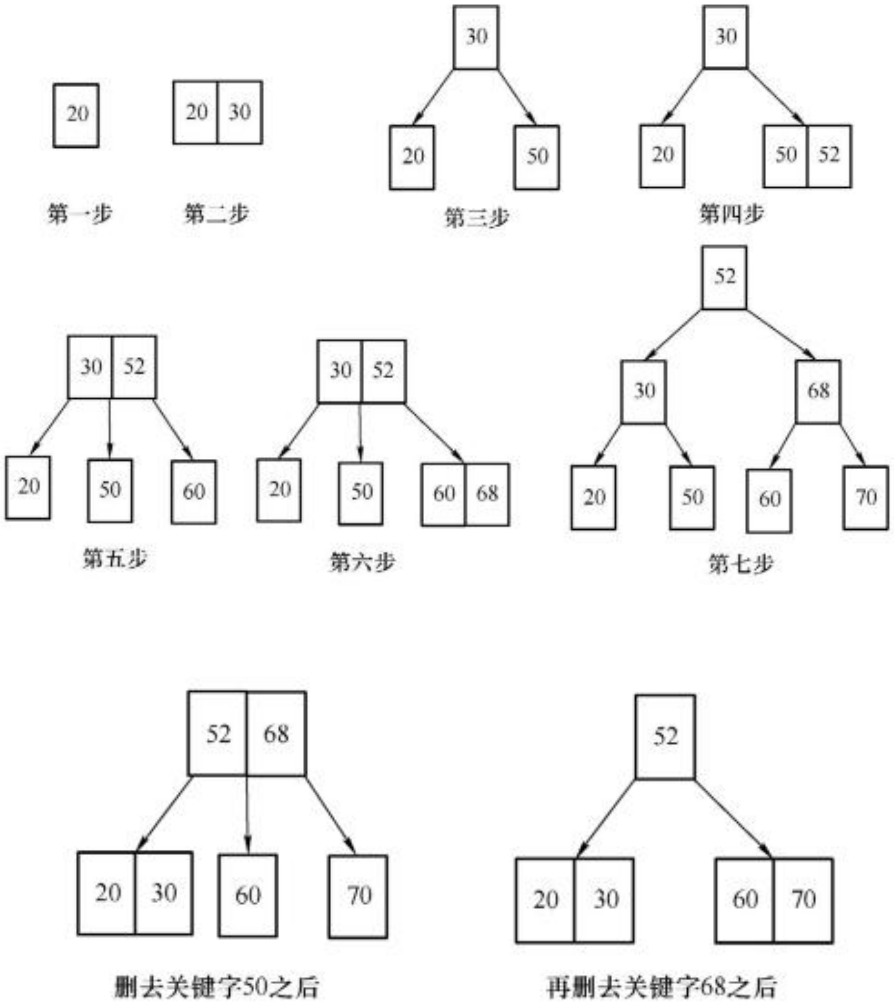
题型分析:这类题没有什么技巧,考生只要始终记住 B-树的概念就行了。就是说,不管是插入还是删除,一定要始终保证 B-树的阶数不变。

8.在如图所示 4 阶 B-树中插入关键字 87,试画出插入调整后树的形状。





9.试从空树开始,画出按以下次序生成3阶B-树的过程:20,30,50,52,60,68,70,如果此后删除50和68,画出每一步执行后的3阶B-树的状态。



四、哈希表的构建

题型分析:这类问题分两步:第一,确定哈希函数;第二,确定处理冲突的方法,然后一步一步地做。需要指出的是,如果哈希函数是除留余数法,则哈希表的表长选择有两种情况:第一,题目规定;第二,如果未规定,则表长就是除数p.

10.若采用链地址法构造散列表,散列函数为H(key)=keyMOD17,则需(　)个链表。这些链的链首指针构成一个指针数组,数组的下标范围为(　)。

A.16,1至17

B.任意,1至16

C.17,0至16

D.13,0至17

11.假设有k个关键字互为同义词,若用线性探测法把这k个关键字存入哈希表中,至少要进行(　)次探测。

A.k-1　　B.K　　C.k+1　　D.k*(k+1)/2

12.设哈希表的表长为14,哈希函数是H(key)=key%11,表中已有数据的关键字15,38,61,84,现将关键字为49的结点加到表中,用二次探测再散列法解决冲突,则放入的位置是(　)。

A.3　　B.5　　C.8　　D.9

五、哈希表的查找过程以及性能分析

题型分析:这类题目有一定的技巧,考生要分清出题人是要求任意哈希表的平均查找长度还是一个具体哈希表的平均查找长度。如果是前者,那么一定与装填因子有关;如果是后者,求平均查,找长度就不能再依靠装填因子了,而要根据具体查找过程来求平均查找长度。

13.采用开放定址法解决冲突的哈希查找中,发生聚集的原因主要是()。

A.数据元素过多

B.负载因子过大

C.哈希函数选择不当

D.解决冲突的算法选择不当

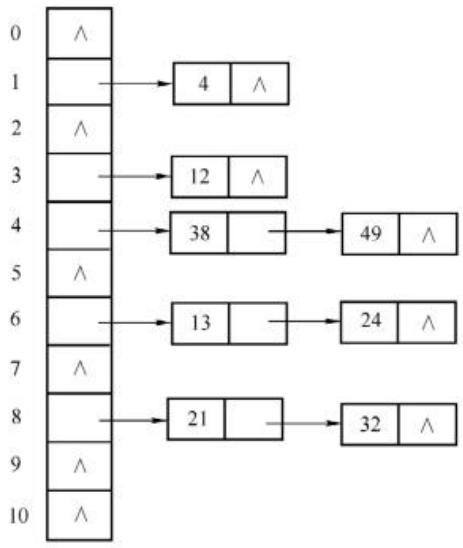
14.设哈希函数H(key)=3*keymod11,哈希地址空间为0到10,对关键字序列{32,13,49,24,38,21,4,12}按下述两种方法解决冲突的方法构造哈希表。

(1)线性探测再散列;

(2)链地址法,并分别求出这两种方法在等概率下查找成功和查找失败时的平均查找长度ASLsucc和ASLunsucc。

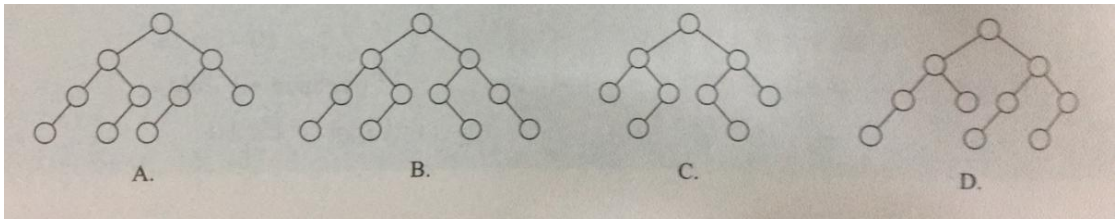
构造的哈希表如下:

哈希地址	0	1	2	3	4	5	6	7	8	9	10
关键字		4		12	49	38	13	24	32	21	
比较次数		1		1	1	2	1	2	1	2	



真题

1.下列二叉树中，可能成为折半查找判定树（不含外部节点）的是（）.



2.下列应用中，适合使用 B+树的是（　）.

A.编译器中的词法分析

B.关系数据库系统中的索引

C.网络中的路由表快速查找

D.操作系统的磁盘空闲块管理

3.在有 n（n≥1000）个元素的升序数组 A 中查找关键字 x，查找算法的伪代码如下所示

k=0;

while（k<n 且 A[k]<x）k=k+3;

if（k<n 且 A[k]==x）查找成功;

elseif（k－1<n 且 A[k－1]==x）查找成功;

elseif（k－2<n 且 A[k－2]==x）查找成功;

else 查找失败;

本算法与折半查找算法相比，有可能具有更少比较次数的情形是（　）。

A.当 x 不在数组中　　B.当 x 接近数组开头处

C.当 x 接近数组结尾处　　D.当 x 位于数组中间位置

4.B+树不同于 B 树的特点之一是（　）.

A.能支持顺序查找

B.节点中含有关键字

C.根节点至少有两个分支

D.所有叶节点都在同一层中

5.下列选项中，不能构成折半查找中关键字比较序列的是（）.

A.500，200，450，180

B.500，450，200，180

C.180，500，200，450

D.180，200，500，450

6.用哈希（散列）方法处理冲突（碰撞）时可能出现堆积（聚集）现象，下列选项中，回收堆积现象影响的是(　).

A.存储效率　　B.散列函数

C.装填（装载）因子　　D.平均查找长度

7.在一棵具有 15 个关键字的 4 阶 B 树中，含有关键字的节点个数最多是（　）.

A.5　　B.6　　C.10　　D.15

8.为提高散列（Hash）表的查找效率，可以采用的正确措施是（　）.

Ⅰ.增大装填（载）因子

Ⅱ.设计冲突（碰撞）少的散列函数

Ⅲ.处理冲突（碰撞）时避免产生聚集（堆积）现象

A.仅Ⅰ　　B.仅Ⅱ

C.仅Ⅰ、Ⅱ　　D.仅Ⅱ、Ⅲ

9.已知一个长度为 16 的顺序表 L，其元素按关键字有序排列。若采用折半查找法查找一个 L 中不存在的元素，则关键字的比较次数最对的是（）.

A.4　　B.5　　C.6　　D.7

10.设包含 4 个数据元素的集合 S={“do”,“for”,“repeat”,“while”}，个元素的查找概率依次为：p1=0.35，p2=0.15，p3=0.15，p4=0.35。将 S 保存在一个长度为 4 的顺序表中，采用折半查找法，查找成功时的平均查找长度为 2.2。请回答：

（1）若用顺序存储结构保存 S，且要求平均查找长度更短，则元素应如何排序？应适用何种查找方法？查找成功时的平均查找长度是多少？

（2）若采用链式存储结构保存 S，且要求平均查找长度更短，则元素应如何排序？应适用何种查找方法？查找成功时的平均查找长度是多少？

更多典型题

1.若查找每个记录的概率均等,则在具有 n 个记录的连续顺序文件中采用顺序查找法查找一个记录，其平均查找长度 ASL 为（　）。

A.（n-1）/2　　B.n/2　　C.（n+1）/2　　D.n

2.顺序查找法适用于查找顺序存储或链式存储的线性表，平均比较次数为（(1)　），二分法查找只适用于查找顺序存储的有序表，平均比较次数为（(2)　）。在此假定 N 为线性表中结点数，且每次查找都是成功的。

A.N+1　　B.$2\log_2 N$　　C.$\log_2 N$　　D.N/2　　E.$N\log_2 N$　　F.N2

3.适用于折半查找的表的存储方式及元素排列要求为（　）。

A.链接方式存储，元素无序

B.链接方式存储，元素有序

C.顺序方式存储，元素无序

D.顺序方式存储，元素有序

4.具有 12 个关键字的有序表，折半查找的平均查找长度为（　）。

A.3.1　　B.4　　C.2.5　　D.5

5.折半查找的时间复杂性为（　）。

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

6.当采用分块查找时，数据的组织方式为（　）。

A.数据分成若干块，每块内数据有序

B.数据分成若干块，每块内数据不必有序，但块间必须有序，每块内最大（或最小）的数据组成索引块

C.数据分成若干块，每块内数据有序，每块内最大（或最小）的数据组成索引块

D.数据分成若干块，每块（除最后一块外）中数据个数需相同

7.下面函数的功能是实现分块查找，空白处应该添加的内容是（　）。

```
int BlkSearch(int * nz,int key,int block,int BLK,int len)
{
    int i;
    block=block-1;
    if(len<=0)
    {
        puts("表为空!");
        return 0;
    }
    if(BLK>len)BLK=len;
    for(i=block * BLK;i<(block+1) * BLK&&nz[i]!=0;i++)
    {
        if(_____)
        {
            printf("找到第%d个数是%d\n",i,key);
            return 0;
        }
    }
    printf("\n");
    printf("查找结束\n");
    return 0;
}
```

A. nz[i]==key　　B. nz[i]==BLK　　C. nz[i]==block　　D. nz[i]==0

8.二叉查找树的查找效率与二叉树的（(1)）有关，在（(2)）时其查找效率最低。

（1）A.高度　　B.结点的多少　　C.树形　　D.结点的位置

（2）A.结点太多　　B.完全二叉树　　C.呈单枝树　　D.结点太复杂

9.在一棵高度为 h 的理想平衡二叉树中，最少含有（　）个结点，最多含有（　）个结点。

A. 2^h 2^{h-1}

B. 2^h-1 2^h

C. 2^h+1 2^h-1

D. 2^{h-1} 2^h-1

10.分别以下列序列构造二叉排序树，与用其他三个序列所构造的结果不同的是（　）。

A.（100，80，90，60，120，110，130）

B.（100，120，110，130，80，60，90）

C.（100，60，80，90，120，110，130）

D.（100，80，60，90，120，130，110）

11.关于 B-树，下列说法中不正确的是（　）。

A.B-树是一种查找树

B.所有的叶结点具有相同的高度

C.2-3 树中，所有非叶子结点有 1 或者 3 个孩子结点

D.通常情况下，B-树不是二叉树

12.在含有 15 个结点的平衡二叉树上，查找关键字为 28（存在该结点）的结点，则依次比较的关键字有可能是（　）。

A.30，36

B.38，48，28

C.48，18，38，28

D.60，30，50，40，38，36

13.下面关于 m 阶 B 树的说法中，正确的是（　）。

①每个结点至少有两棵非空子树。

②树中每个结点至多有 m-1 个关键字。

③所有叶子在同一层上。

④当插入一个数据项引起 B 树结点分裂后，树长高一层。

A.①②③　　B.②③　　C.②③④　　D.③

14.下面关于 B 和 B+树的叙述中，不正确的是（　）。

A.B 树和 B+树都是平衡的多叉树

B.B 树和 B+树都可用于文件的索引结构

C.B 树和 B+树都能有效地支持顺序检索

D.B 树和 B+树都能有效地支持随机检索

15.m 阶 B-树是一棵（　）。

A.m 叉排序树

B.m 叉平衡排序树

C.m-1 叉平衡排序树

D.m+1 叉平衡排序树

16.若对有 18 个元素的有序表做二分查找，则查找 A［3］的比较序列的下标为（　）。

A.1，2，3　　B.9，4，2，3　　C.10，5，3　　D.9，2，3

17.有一个有序表｛1，3，9，12，32，41，45，62，75，77，82，95，100｝，当用二分查找法查找值为 82 的结点时，经（　）次比较后查找成功。

A.1　　B.2　　C.4　　D.8

18.采用分块查找时，若线性表中共有 625 个元素，查找每个元素的概率相同，假设采用顺序查找来确定结点所在的块，则每块分为（　）个结点最佳。

A.9　　B.25　　C.6　　D.625

19.在有 n 个结点且为完全二叉树的二叉排序树中查找一个键值，其平均比较次数的数量级为（　）。

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

20.对包含 n 个关键码的散列表进行检索，平均检索长度为（　）。

A.O（$\log_2 n$）　　B.O（n）　　C.O（$n\log_2 n$）　　D.不直接依赖于 n

21.在散列表上，每个地址单元所链接的同义词表的（　）。

A.键值相同　　B.元素值相同　　C.散列地址相同　　D.含义相同

22.设哈希表长 m=14，哈希函数 H（key）=keymod11。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法处理冲突，则关键字为 49 的结点的地址是（　）。

A.8　　B.3　　C.5　　D.9

综合题

1.请编写一个判别给定二叉树是否为二叉排序树的算法，设二叉树用 llink-rlink 法存储。

2.某个任务的数据模型可以抽象为给定的 k 个集合：S1，S2，…，Sk。其中 Si（1≤i≤k）中的元素个数不定。在处理数据过程中将会涉及元素的查找和新元素的插入两种操作，查找和插入时用一个二元组（i，x）来规定一个元素，i 是集合的序号，x 是元素值。设计一种恰当的数据结构来存储这 k 个集合的元素，并能高效地实现所要求的查找和插入操作。

（1）构造数据结构，并且说明选择的理由。

（2）若一组数据模型为 S1=｛10.2，1.7，4.8，16.2｝，S2=｛1.7，8.4，0.5｝，S3=｛4.8，4.2，3.6，2.7，5.1，3.9｝，待插入的元素二元组为（2，11.2）和（1，5.3），按你的设计思想画出插入元素前后的数据结构状态。

3.假设 K1，…，Kn 是 n 个关键词，试解答：

（1）试用二叉查找树的插入算法建立一棵二叉查找树，即当关键词的插入次序为 K1，K2，…，Kn 时，用算法建立一棵以 LLINK-RLINK 链接表示的二叉查找树。

（2）设计一个算法，打印出该二叉查找树的嵌套括号表示结构。假定该二叉查找树的嵌套括号表示结构为 B（A，D（C，E））。

4.写出在二叉排序树中删除一个结点的算法，使删除后仍为二叉排序树。设删除结点由指针 p 所指，其双亲结点由指针 f 所指，并假设被删除结点是其双亲结点的右孩子。描述上述算法。

5.已知二叉树排序树中某结点指针 p，其双亲结点指针为 fp，p 为 fp 的左孩子。试编写算法，删除 p 所指结点。

6.二叉排序树采用二叉链表存储。写一个算法，删除结点值是 X 的结点。要求删除该结点后，此树仍然是一棵二叉排序树，并且高度没有增长(注意:可不考虑被删除的结点是根的情况)。

7.设记录 R1，R2，…，Rn 按关键字值从小到大顺序存储在数组 r［1..n］中，在 r［n+1］处设立一个监督哨，其关键字值为+∞。试写一查找给定关键字 k 的算法，并画出此查找过程的判定树，求出在等概率情况下查找成功时的平均查找长度。

8.给出折半查找的递归算法，并给出算法时间复杂度分析。

第六章　排序

排序　大纲要求

- 各种排序算法的基本原理（☆☆☆）
- 各种排序算法的应用（☆☆）

一、基本概念

- 排序基本概念
- 常见查找算法基本内容，特点，对比

1.排序方法的稳定性是指(　)。

A.该排序算法允许有相同的关键字记录

B.平均时间为0(nlogn)的排序方法

C.该排序算法不允许有相同的关键字记录

D.以上说法均不对

2.下列说法中正确的是(　)。

A.不管初始数据如何,快速排序永远都不会比气泡排序慢

B.折半插入排序仅减少了关键字间的比较次数,而记录的移动次数不变

C.直接插入排序、气泡排序、简单选择排序都是基于“交换”操作的排序方法

D.内部排序就是在计算机内部进行的排序

3.以下序列不是堆的是(　)。

A.(100,85,98,77,80,60,82,40,20,10,66)

B.(100,98,85,82,80,77,66,60,40,20,10)

C.(10,20,40,60,66,77,80,82,85,98,100)

D.(100,85,40,77,80,60,66,98,82,10,20)

二、各种排序算法的排序过程

题型分析:考生要根据考纲对各种排序方法进行掌握,具体地说,就是能够用自己的双手模出计算机的执行过程即可。

4.对下列关键字用快速排序方法进行排序时,速度最快的是(　)。

A.(21,25,5,17,9,23,30)

B.(25,23,30,17,21,5,9)

C.(21,9,17,30,25,23,5)

D.(5,9,17,21,23,25,30)

5.对序列(15,9,7,8,20,-1,4)用希尔排序方法排序,经过一趟排序后,序列变,为(15,-1,4,8,20,9,7),则该次采用的增量是(　)。

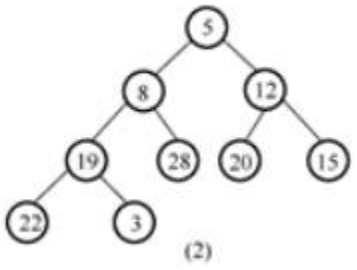
A.1　　B.4　　C.3　　D.2

6.采用递归方式对顺序表进行快速排序,下列关于递归次数的叙述中正确的是(　)。

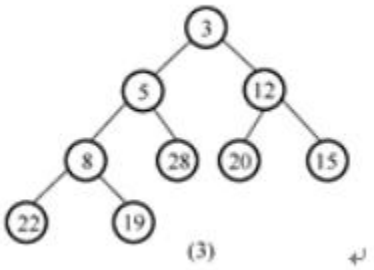
A.递归次数与初始数据的排列次序无关

B. 每次划分后,先处理较长的分区可以减少递归次数
C. 每次划分后,先处理较短的分区可以减少递归次数
D. 递归次数与每次划分后得到的分区的处理顺序无关

7. 已知关键字序列(5,8,12,19,28,20,15,22)是小根堆,插入关键字 3,调整,后得到的小根堆是(　)。



(2)



(3)

A. (3,5,12,8,28,20,15,22,19)
B. (3,5,12,19,20,15,22,8,28)
C. (3,8,12,5,20,15,22,28,19)
D. (3,12,5,8,28,20,15,22,19)

8. 以关键字序列(503,087,512,061,908,170,897,275,653,426)为例,手工模拟简单选择排序算法、归并排序算法、基数排序算法,写出每一趟排序结束时的关键字状态。

1）简单选择排序

原始序列：503，087，512，061，908，170，897，275，653，426

第一趟：　061，087，512，503，908，170，897，275，653，426

第二趟：　061，087，512，503，908，170，897，275，653，426

第三趟：　061，087，170，503，908，512，897，275，653，426

第四趟：　061，087，170，275，908，512，897，503，653，426

第五趟：　061，087，170，275，426，512，897，503，653，908

第六趟：　061，087，170，275，426，503，897，512，653，908

第七趟：　061，087，170，275，426，503，512，897，653，908

第八趟：　061，087，170，275，426，503，512，653，897，908

第九趟：　061，087，170，275，426，503，512，653，897，908

2）归并排序

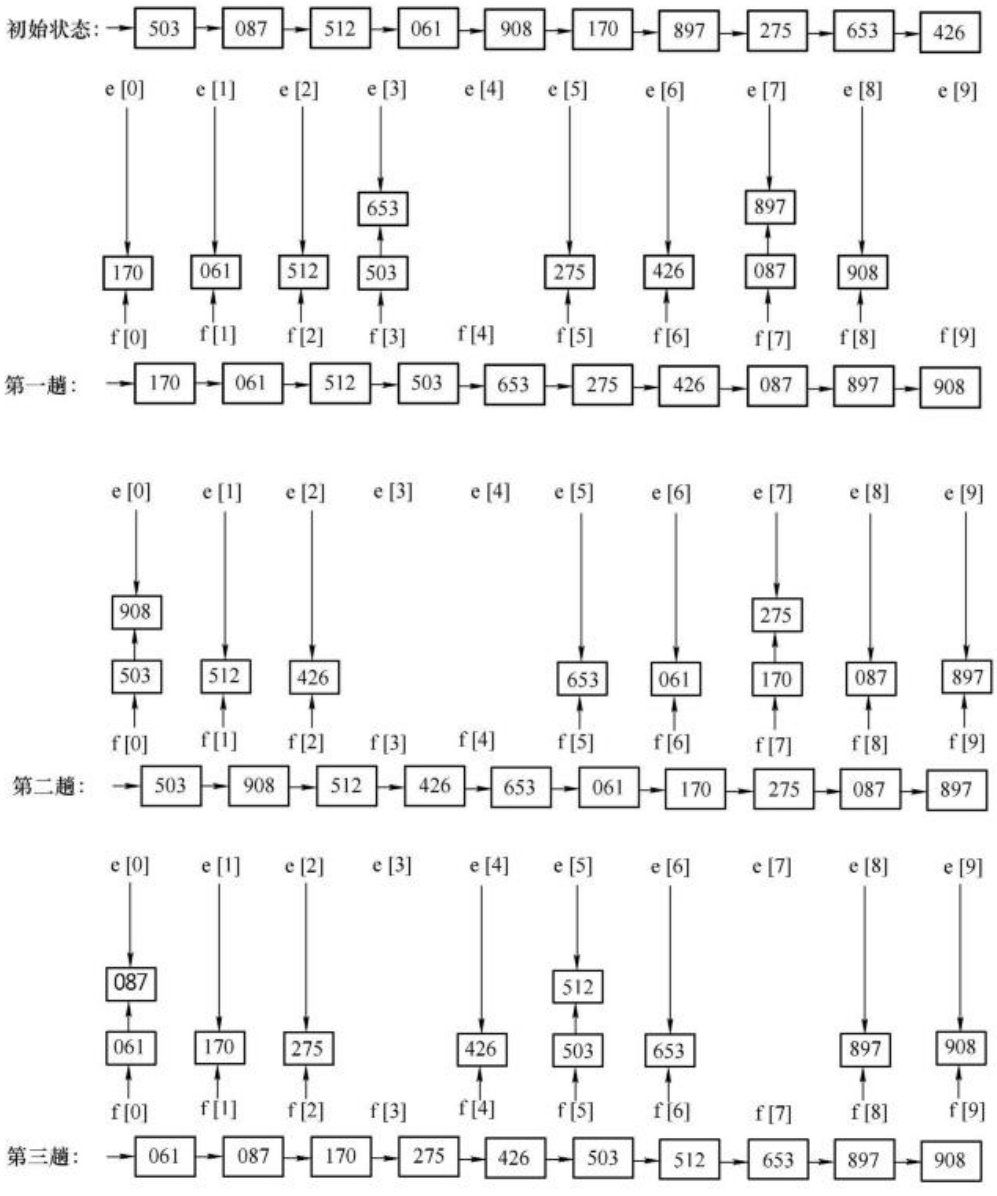
原始序列：503，087，512，061，908，170，897，275，653，426

第一趟： 087，503，061，512，170，908，275，897，426，653

第二趟： 061，087，503，512，170，275，897，908，426，653

第三趟： 061，087，170，275，503，512，897，908，426，653

第四趟： 061，087，170，275，426，503，512，653，897，908



三、各种排序算法的效率分析

题型分析:做这类题之前,考生应明白考查一种排序要从三方面着手:时间、空间、稳定性。大家还应牢记一条规律:时间、空间、稳定性这三方面往往不可兼得。

9.就平均性能而言,目前最好的内部排序方法是(　)。

A.气泡排序

B.希尔排序

C.插入排序

D.快速排序

10.就排序算法所用的辅助空间而言,堆排序、快速排序、归并排序的关系是(　)。

A.堆排序<快速排序<归并排序

B.堆排序<归并排序<快速排序

C.堆排序>归并排序>快速排序

D.堆排序>快速排序>归并排序

11.直接插入排序在最好情况下的时间复杂度为(　)。

A.$O(\log_2 n)$

B.$O(n)$

C.$O(n\log_2 n)$

D.$O(n^2)$

四、各种排序方法的稳定性

题型分析:一般说来,效率比较低的排序方法是稳定的,而效率高的是不稳定的,但是归并排序、基数排序是稳定的。

12.若要尽可能地完成对实数数组的排序,且要求排序是稳定的,则应选择(　)。

A.快速排序

B.堆排序

C.归并排序

D.基数排序

13.若要求在$O(n\log_2 n)$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.直接插入排序

D.归并排序

五、排序方法的应用

题型分析:做这类题要分两步走。第一,了解题意;第二,根据所学的方法找出最好的解决方案。另外,如果题目要求考生分析一个新的排序方法,则要从时间、空间、稳定性方面着手分析。

14.将两个各有 N 个元素的有序表归并成一个有序表,其最少的比较次数是(　)。

A.N-1　　B.N　　C.2N-1　　D.2N.

15.有一种简单的排序算法,称为计数排序。这种排序算法对一个待排序的表(用数组表示)进行排序,并将排序结果存放到一个新的表中。需要注意的是,该表中所有待排序的关键字互不相同,计数排序算法针对表中的每个记录,扫描待排序的表一趟,统计表中有多少个记录的关键字比该关键字小,假设对某一个记录统计出数值为 c,那么这个记录在新的有序表中的合适的存放位置即为 c。

(1)给出适用于计数排序的数据表的定义。

(2)编写实现计数排序的算法。

(3)对于有 n 个记录的表,比较次数是多少?

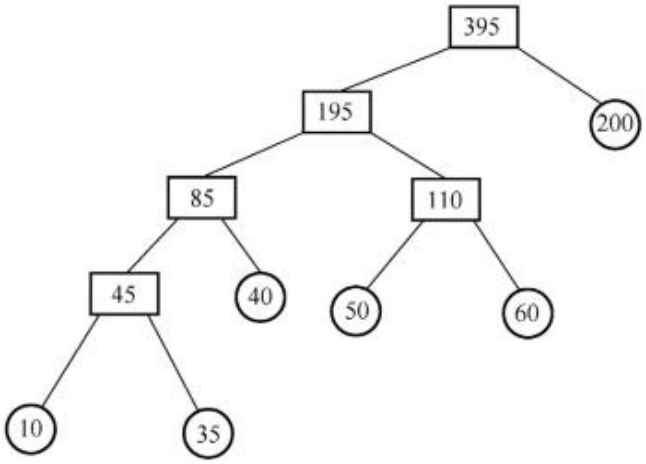
(4)与直接选择排序相比,这种方法是否更好?为什么?

16.设有 6 个有序表 A,B，C，D，E，F,分别含有 10、35、40、50、60 和 200 个数据元素,各表中元素按升序排列。要求通过 5 次两两合并,将最终合并成一个升序表,并在最坏情况下比较的总次数达到最小。

请回答下列问题:

(1)给出完整的合并过程,并求出最坏情况下的比较总次数。

(2)根据你的合并过程,描述 n(n>2)个不等长升序表的合并策略,并说明理由。



真题

1.在内部排序时，若选择了归并排序而没有选择插入排序，这可能的理由是（）.

Ⅰ.归并排序的程序代码更短

Ⅱ.归并排序的占用空间更小

Ⅲ.归并排序的运行效率更高

A.仅Ⅱ　　B.仅Ⅲ

C.仅Ⅰ、Ⅱ　　D.仅Ⅰ、Ⅲ

2.下列排序方法中，若将顺序存储更换为链式存储，则算法的时间效率会降低的是（）.

Ⅰ.插入排序

Ⅱ.选择排序

Ⅲ.起泡排序

Ⅳ.希尔排序

Ⅴ.堆排序

A.仅Ⅰ、Ⅱ　　B.仅Ⅱ、Ⅲ

C.仅Ⅲ、Ⅳ　　D.仅Ⅳ、Ⅴ

3.对 10TB 的数据文件进行排序，应使用的方法是（）.

A.希尔排序　　B.堆排序

C.快速排序　　D.归并排序

4.已知小根堆为 8，15，10，21，34，16，12，删除关键字 8 之后需重建堆，在此过程中，关键字之间的比较次数是（）.

A.1　　B.2　　BC.3　　BD.4

5.希尔排序的组内排序采用的是（）.

A.直接插入排序

B.折半插入排序

C.快速排序

D.归并排序

6.用希尔排序方法对一个数据序列进行排序时，若第 1 趟排序结果为 9，1，4，13，7，8，20，23，15，则该趟排序采用的增量（间隔）可能是（）.

A.2　　BB.3　　BC.4　　BD.5

7.下列选项中，不可能是快速排序第 2 趟排序结果的是（）.

A.2，3，5，4，6，7，9

B.2，7，5，6，4，3，9

C.3，2，5，4，7，6，9

D.4，2，3，5，7，6，9

8.对给定的关键字序列 110，119，007，911，114，120，122 进行基数排序，则第 2 趟分配收集后得到的关键字序列是（）.

A.007,110,119,114,911,120,122

B.007,110,119,114,911,122,120

C.007,110,911,114,119,120,122

D.110,120,911,122,114,007,119

9.在内部排序过程中，对尚未确定最终位置的所有元素进行一遍处理称为一趟排序。下列排序方法中，每一趟排序结束时都至少能够确定一个元素最终位置的方法是（）.

Ⅰ.简单选择排序

Ⅱ.希尔排序

Ⅲ.快速排序

Ⅳ.堆排序

Ⅴ.二路归并排序

A.仅Ⅰ、Ⅲ、Ⅳ　　B.仅Ⅰ、Ⅲ、Ⅴ

C.仅Ⅱ、Ⅲ、Ⅳ　　D.仅Ⅲ、Ⅳ、Ⅴ

10.对一待排序列分别进行折半插入排序和直接插入排序，两者之间可能的不同之处是（ ）.

A.排序的总趟数

B.元素的移动次数

C.使用辅助空间的数量

D.元素之间的比较次数

11.为实现快速排序算法，待排序序列宜采用的存储方式是（ ）.

A.顺序存储　　B.散列存储

C.链式存储　　D.索引存储

12.已知序列 25，13，10，12，9 是大根堆，在序列尾部插入新元素 18，将其再调整为大根堆，调整过程中元素之间进行的比较次数是（ ）.

A.1　　B.2　　C.4　　D.5

13.采用递归方式对顺序表进行快速排序。下列关于递归次数的叙述中，正确的是（ ）.

A.递归次数与初始数据的排列次序无关

B.每次划分后，先处理较长的分区可以减少递归次数

C.每次划分后，先处理较短的分区可以减少递归次数

D.递归次数与每次划分后得到的分区的处理顺序无关

14.对{05，46，13，55，94，17，42}进行基数排序，一趟排序的结果是().

A.05，46，13，55，94，17，42

B.05，13，17，42，46，55，94

C.42，13，94，05，55，46，17

D.05，13，46，55，17，42，94

15.下列序列中，（ ）是执行第 1 趟快速排序后得到的序列。

A.[68，11，18，69][23，93，73]

B.[68，11，69，23][18，93，73]

C.[93，73][68，11，69，23，18]

D.[73，11，69，23，18][93，68]

16.在快速排序过程中，下列结论正确的是（）.

A.左、右两个子表都以各自排好序

B.左边的元素都不大于右边的元素

C.左边子表长度小于右边子表长度

D.左、右两边元素的平均值相等

17.已知由 n（n≥2）个正整数构成的集合 A={ak|0≤k<n}，将其划分为两个不相交的子集 A1 和 A2。元素个数分别是 n1 和 n2，A1 和 A2 中元素之和分别为 S1 和 S2。设计一个尽可能高效的划分算法。满足|n1－n2|最小且|S1－S2|最大。要求：

（1）给出算法的基本设计思想。

（2）根据设计思想，采用 C 或 C++语言描述算法，关键之处给出注释。

（3）说明你所设计算法的平均时间复杂度和空间复杂度。

更多典型题

1.下面给出的 4 种排序方法中，（ ）排序法是不稳定性排序法。

A.插入　　B.冒泡　　C.二路归并　　D.堆

2.下列内部排序算法中，其比较次数（或交换次数）与序列初态无关的算法是（ ）。

A.快速排序　　B.直接插入排序　　C.二路归并排序　　D.冒泡排序

3.下列内部排序算法中,在初始序列已基本有序(除去 n 个元素中的某 k 个元素后即呈有序，k＆lt；＆lt；n）的情况下，排序效率最高的算法是（ ）。

A.冒泡排序　B.堆排序　C.直接插入排序　D.二路归并排序

4.下列排序算法中，（ ）每一趟都能选出一个元素放在最终位置上，并且是不稳定的。

A.冒泡排序

B.希尔排序

C.简单选择排序

D.直接插入排序

5.下列排序方法中，时间复杂性不受数据初始状态影响，恒为 O（nlog2n）的是（ ）。

A.堆排序　　B.冒泡排序　　C.直接选择排序　　D.快速排序

6.下列排序算法中，某一趟结束后未必能选出一个元素放在其最终位置上的是（ ）。

A.选择排序　　B.冒泡排序　　C.归并排序　　D.堆排序

7.下列排序算法中，在每一趟都能选出一个元素放到其最终位置上，并且其时间性能受数据初始特性影响的是（　）。

A.直接插入排序　　B.归并排序

C.直接选择排序　　D.堆排序

8.对初始状态为递增序列的表按递增顺序排序，最省时间的是（(1)）算法，最费时间的是（(2)）算法。

A.堆排序　　B.快速排序　　C.插入排序　　D.归并排序

9.如果只想得到1000个元素组成的序列中第5个最小元素之前的部分排序的序列，用（　）方法最快。

A.冒泡排序　　B.快速排序　　C.简单选择排序　　D 堆排序

10.数据表A中有10000个元素，如果仅要求求出其中最大的10个元素，则采用（　）方法最节省时间。

A.堆排序　　B.希尔排序　　C.快速排序　　D.直接选择排序

11.若对序列（tang，deng，an，wang，shi，bai，fang，liu）采用简单选择排序法按字典顺序进行排序，下面给出的四个序列中，第三趟的结果是（　）。

A.an，bai，deng，wang，tang，fang，shi，liu

B.an，bai，deng，wang，shi，tang，fang，liu

C.an，bai，deng，wang，fang，shi，tang，liu

D.an，bai，deng，wang，shi，liu，tang，fang

12.从未排序序列中依次取出一个元素与已排序序列中的元素依次进行比较，然后将其放在已排序序列的合适位置，该排序方法称为（　）排序法。

A.插入　　B.选择　　C.希尔　　D.二路归并

13.若用冒泡排序方法对序列{10，14，26，29，41，52}从大到小排序，需进行（　）次比较。

A.3　　B.10　　C.15　　D.25

14.下列（　）是一个堆。

A.19，75，34，26，97，56

B.97，26，34，75，19，56

C.19，56，26，97，34，75

D.19，34，26，97，56，75

15.若一组记录的关键码为（46，79，56，38，40，84），则利用快速排序的方法，以第1个记录为基准得到的一次划分结果为（　）。

A.38，40，46，56，79，84

B.40，38，46，79，56，84

C.40，38，46，56，79，84

D.40，38，46，84，56，79

16.采用简单选择排序，比较次数与移动次数分别为（）

A.O（n），O（$\log_2 n$）　　B.O（$\log_2 n$），O（n^2）

C.O（n^2），O（n）　　D.O（$n\log_2 n$），O（n）

17.就排序算法所用的辅助空间而言，堆排序、快速排序、归并排序的关系是（　）。

A.堆排序＆lt；快速排序＆lt；归并排序

B.堆排序＆lt；归并排序＆lt；快速排序

C.堆排序＆gt；归并排序＆gt；快速排序

D.堆排序＆gt；快速排序＆gt；归并排序

18.一组记录的关键码为（25，48，16，35，79，82，23，40，36，72），其中，含有5个长度为2的有序表，按归并排序的方法对该序列进行一趟归并后的结果为（　）。

A.16，25，35，48，23，40，79，82，36，72

B.16，25，35，48，79，82，23，36，40，72

C.16，25，48，35，79，82，23，36，40，72

D.16，25，35，48，79，23，36，40，72，82

19.已知10个数据元素为（54，28，16，34，73，62，95，60，26，43），对该序列按从小到大排序，经过一趟冒泡排序后的序列为（　）。

A.16，28，34，54，73，62，60，26，43，95

B.28，16，34，54，62，73，60，26，43，95

C.28，16，34，54，62，60，73，26，43，95

D.16，28，34，54，62，60，73，26，43，95

20.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

其所采用的排序方法是（　）。

A.直接选择排序　　B.希尔排序　　C.归并排序　　D.快速排序

21.在对一组记录（50，40，95，20，15，70，60，45，80）进行直接插入排序时，当把第7个记录60插入到有序表时，为寻找插入位置需比较（　）次。

A.1　　B.2　　C.3　　D.4

22.将两个各有N个元素的有序表归并成一个有序表，其最少的比较次数是（　）。

A.N　　B.2N-1　　C.2N　　D.N-1

23.已知待排序的n个元素可分为n/k个组，每个组包含k个元素，且任一组内的各元素均分别大于前一组内的所有元素和小于后一组内的所有元素，若采用基于比较的排序，其时间下界应为（　）。

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

24.已知关键序列5，8，12，19，28，20，15，22是小根堆（最小堆），插入关键字3，调

整后得到的小根堆是（　）。

A.3，5，12，8，28，20，15，22，19

B.3，5，12，19，20，15，22，8，28

C.3，8，12，5，20，15，22，28，19

D.3，12，5，8，28，20，15，22，19

25.归并排序中，归并的趟数是（　）。

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

26.有一组数据（15，9，7，8，20，-1，7，4），用堆排序的筛选方法建立的初始堆为（　）。

A.-1，4，8，9，20，7，15，7

B.-1，7，15，7，4，8，20，9

C.-1，4，7，8，20，15，7，9

D.A、B、C 均不对

27.基于比较方法的 n 个数据的内部排序，最坏情况下情况下的时间复杂度能达到的最好下界是（　）。

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

28.以下排序方法中，稳定的排序方法是（　）。

A.直接插入排序　　B.直接选择排序　　C.堆排序　　D.基数排序

29.在对一组记录（50，40，95，20，15，70，60，45，80）进行希尔排序时，假定 d0=9，d1=4，d2=2，d3=1，则第二趟排序结束后前 4 条记录为（　）。

A.（50，20，15，70）

B.（60，45，80，50）

C.（15，20，50，40）

D.（15，20，80，70）

30.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，在第三趟归并中，是把长度为（　）的有序表归并为长度为（　）的有序表。

A.5，4，8　　B.6，3，9　　C.7，4，3　　D.3，8，2