数据结构复习题 2020秋季

1. **绪论**

一、选择题

1、把数据存储到计算机中，并具体体现数据元素间的逻辑结构称为（ ）。

A. 给相关变量分配存储单元

B. 物理结构

C. 算法的具体实现

D. 逻辑结构

2、下列说法中，不正确的是（ ）。

A. 数据项是数据中不可分割的最小可标识单位

B. 数据元素是数据的基本单位

C. 数据可有若干个数据元素构成

D. 数据项可由若干个数据元素构成

3、一个存储结点存储一个（ ）。

A. 数据元素

B. 数据结构

C. 数据项

D. 数据类型

4、数据结构中，与所使用的计算机无关的是数据的（ ）。

A. 存储结构

B. 物理和存储结构

C. 物理结构

D. 逻辑结构

5、下列的叙述中，不属于算法特性的是（ ）。

A. 可行性

B. 输入性

C. 可读性

D. 有穷性

6、算法的时间复杂度与（ ）有关。

A. 计算机的操作系统

B. 算法本身

C. 数据结构

D. 所使用的计算机。

7、下面程序段的时间复杂度是（ ）。

i=s=0;

while (s<n){

i++;

s+=i;

}

A.O(n0.5) B. O(log2n) C. O(n) D.O(1)

8、下面程序段的时间复杂度是（ ）。

int f(unsigned int n){

if (n==0||n==1) return 1;

else return n\*f(n-1);

}

A.O(1) B. O(log2n) C. O(n!) D. O(n)

10、在数据结构中，从逻辑上可以把数据结构分为（ ）。

A.动态结构和静态结构 B.紧凑结构和非紧凑结构

C.内部结构和外部结构 D.线性结构和非线性结构

11、执行下面程序段时，执行S语句的次数为（ ）。

for (int i=1;i<=n;i++)

for (int j=1;i<=i;j++)

S;

A.n2 B.n2/2 C.n(n+1) D.n(n+1)/2

12、数据的存储结构包括数据元素的表示和（ ）。

A. 数据元素间的关系的表示

B. 数据处理的方法

C. 数据元素的类型

D. 相关算法

13、树状结构中数据元素的位置之间存在（ ）的关系。

A. 一对一

B. 多对多

C. 每一个元素都有一个直接前驱和一个直接后继

D. 一对多

14、一种逻辑结构（ ）。

A. 与存储该逻辑结构的计算机相关

B. 只能有唯一的存储结构

C. 可以有不同的存储结构

D. 是指某一种数据元素的性质

15、把数据存储到计算机中，并具体体现数据元素间的逻辑结构称为（ ）。

A. 逻辑结构

B. 数据元素的存储

C. 存储结构

D. 给数据元素分配存储空间

二、判断题

1. 数据元素是数据的最小单位。（ ）

2. 数据的逻辑结构是指各数据元素之间的逻辑关系，是用户根据应用需要建立的。（ ）

3. 算法和程序原则上没有区别，在讨论数据结构时二者是通用的。（ ）

4. 数据的逻辑结构与数据元素本身的内容和形式无关。（ ）

5. 算法和程序都应具有下面一些特征：有输入，有输出，确定性，有穷性，有效性。（ ）

6. 只有用面向对象的计算机语言才能描述数据结构算法。（ ）

7. 数据元素可以有一个或多个数据项组成。（ ）

8. 数据元素之间的抽象关系称为物理结构。（ ）

9. 数据的逻辑结构在计算机中的表示称为逻辑结构。（ ）

10. 数据的逻辑结构是与存储该结构的计算机相关的。（ ）

11．数据结构中,元素之间存在多对多的关系称为树状结构。（ ）

12. 通常可以把某城市中各公交站点间的线路图抽象成树型结构。（ ）

13. 通常可以把一本含有不同章节的书的目录结构抽象成线性结构。（ ）

14. 结构中的数据元素存在多对多的关系称为图形结构。（ ）

15．数据结构中，数据可以由一个或多个数据项组成。（ ）

三、程序题

指出下列各算法的时间复杂度。

1、int prime(int n)

{

int i=1;

int x=(int) sqrt(n);

while (++i<=x)

if (n %i= =0) break;

if (i>x) return 1;

else return 0;

}

2、int suml(int n)

{

int p=1,s=0;

for (int i=1; i<=n; i++) {

p\*=i;

s+=p;

}

return s;

}

3、 int sum2(int n)

{

int s=0;

for (int I=1; I<=n; I++) {

int p=1;

for (int j=1; j<=I; j++)

p\*=j;

s+=p;

}

return s;

}

4、 int fun (int n)

{

int I=1, s=1;

while (s<n)

s+=++I;

return I;

}

**第二章 线性表**

一、选择题

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），插入一个元素，则移动元素个数为（ ）。

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

2、设有一个长度为n的顺序表，要删除第i个元素移动元素的个数为（ ）。A. I B. n-i-1 C. n-i D. n-i+1

3、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，可用语句（ ）。

A. q->next=NULL B. p->next=q->next

C. p=q->next D. p->next=q

4、在一个单链表中p所指结点之后插入一个s所指的结点时，可执行（ ）。

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

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

5、非空的单向循环链表的尾结点满足（ ）（设头指针为head，指针p指向尾结点）。

A. p->next==head B. p==NULL

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

6、链表不具有的特点是（ ）。

A. 不必事先估计存储空间

B. 可随机访问任一元素

C. 逻辑上相邻的元素在物理位置上不一定相邻

D. 插入删除不需要移动元素

7、带头结点的链表为空的判断条件是（ ）（设头指针为head）。

A. head->next==head

B. head ==NULL

C. head->next==NULL

D. head!=NULL

8、在一个长度为n的顺序表中为了删除第5个元素，由第6个元素开始从后到前依次移动了15个元素。则原顺序表的长度为（ ）。

A. 19 B. 21 C. 20 D. 25

9、有关线性表的正确说法是（ ）。

A. 线性表至少要求一个元素

B. 每个元素都有一个直接前驱和一个直接后继

C. 表中的元素必须按由小到大或由大到下排序

D. 除了一个和最后一个元素外，其余元素都有一个且仅有一个直接前驱和一个直接后继

10、向一个有127个元素的顺序表中插入一个新元素，并保持原来的顺序不变，平均要移动（ ）个元素。

A. 63.5 B. 7

C. 63 D. 8

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

A. 106 B. 98

C. 102 D. 100

12、在一个不带头结点的单循环链表中，p、q分别指向表中第一个结点和尾结点，现要删除第一个结点，且p、q仍然分别指向新表中第一个结点和尾结点。可用的语句是p=p->next;和（ ）。

A. q->next=p B. p->next=q

C. q=p D. p=q->next

13、在线性表的顺序结构中，以下说法正确的是（ ）。

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. 进行数据元素的插入、删除效率较高

D. 逻辑上相邻的元素在物理位置上也相邻

14、对链表, 以下叙述中正确的是（ ）。

A. 不能随机访问任一结点

B. 插入删除元素的操作一定要要移动结点

C. 结点占用的存储空间是连续的

D. 可以通过下标对链表进行直接访问

15、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），插入一个元素，则移动元素个数为（ ）。

A. n-i+1 B. i

C. n-i-1 D. n-i

16、在一个单链表HL中，若要向表头插入一个由指针p指向的结点，则执行\_\_\_\_\_\_\_\_\_。

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

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

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

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

17、在一个单链表中，已知q所指结点是p所指结点的前驱结点，若在q和p之间插入s结点，则以下操作哪个是正确的（ ）。

A. s->next=p->next；p->next=s；

B. p->next=s->next；s->next=p；

C. p->next=s；s->next=q；

D. q->next=s；s->next=p；

18、在循环双链表的p所指结点之后插入s所指结点的操作是（ ）。

A.p->right=s；s->left=p；p->right->left=s；s->right=p->right；

B. p->right=s；p->right->left=s；s->left=p；s->right=p->right；

C. s->left=p；s->right=p->right；p->right=s；p->right->left=s；

D. s->left=p；s->right=p->right；p->right->left=s；p->right=s；

19、若HL为一个不带表头结点的循环单链表的表头指针，若有HL->next= =HL条件存在，则该循环单链表是（ ）。

A.空表 B.只有1个元素；

C.空表或只有一个元素 D.非空表

20、若HL为一个带表头结点的单链表的表头指针，则该表为空表的条件是（ ）。

A.HL==NULL B.HL->next==NULL

C.HL->next==HL D.HL!=NULL

21、设头指针为head的非空的单向链表，指针p指向尾结点,则通过以下操作（ ）可使其成为单向循环链表。

选择一项：

A. head = p; B. p=head;

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

二、判断题

1、 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;。（ ）

2、 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 。（ ）

3、 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p->next==head;的结果为真，则p所指结点为尾结点。（ ）

4、 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 p->next=s; s->next= p->next；的操作。（ ）5、 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。（ ）

6、 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；。（ ）

7、 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；。（ ）

8、 设有一个长度为40的顺序表，要删除第8个元素需移动元素的个数为3。（ ）

9、 线性表用关键字的顺序方式存储，可以用二分法排序。（ ）

10、线性表用顺序方式存储可以随机访问。（ ）

三、程序填空

1、设线性表以不带头结点的单向链表存储，链表头指针为head，以下程序的功能是输出链表中各结点中的数据域data,完成程序中空格部分。

#define NULL 0

void main（ ）

{ NODE \*head ,\*p ;

p=head; /\*p为工作指针\*/

do

{printf(“%d\n”,

[ 1 ];

[ 2 ];

}while

[ 3 ]；

2、 设有一个头指针为head的不带头结点单向链表，p、q是指向链表中结点类型的指针变量，p指向链表中结点a, （设链表中没有结点的数据域与结点a的数据域相同）,写出相关语句

(1)使该单向链表成为单向循环链表

(2)插入结点s,使它成为a结点的直接前驱

q=p; x=p->data;

while（[ 4 ]）q=q->next;

q->next=head;

q=p; p=p->next;

while(p->data!=x)

{ q=p;

[ 5 ]

}

s->next=p;

[ 6 ]

3、设有一个不带头结点的单向链表，头指针为head，p、prep是指向结点类型的指针，该链表在输入信息时不慎把相邻两个结点的信息重复输入，以下程序段是在该单向链表中查找这相邻两个结点，把该结点的数据域data打印出来，并把其中之一从链表中删除，填写程序中的空格。

prep=head;

p=prep->next;

while(p->data!=prep->data)

{

prep=p;

[ 7 ];

}

printf(“min=%d”,

[ 8 ]);

prep->next= [ 9 ];

**第三章 数组和广义表**

一、选择题

1．若让元素1，2，3依次进栈，则出栈顺序不可能为（ ）。

A．3，2，1 B．2，1，3

C．3，1，2 D．1，3，2

2．一个队列的入队序列是1，2，3，4。则队列的输出序列是（ ）。

A．4，3，2，1 B．1，2，3，4

C．1，4，3，2 D．3，2，4，1

3. 一个栈的进栈序列是10，20，30，40，50，则栈的不可能输出序列是（ ）（进栈出栈可以交替进行）。

A．10,20,30,40,50 B．40,30,50,10,20

C．40,50,30,20,10 D．50,40,30,20,10

4. 元素4，6，8，10按顺序依次进栈，按该栈的可能输出序列依次入队列，该队列的可能输出序列是（ ）（进栈出栈可以交替进行）。

A．10，8，4，6 B．10，6，4，8

C．8，4，6，10 D．10，8，6，4

5．向顺序栈中压入新元素时，应当（ ）。

A．先移动栈顶指针，再存入元素

B．先存入元素，再移动栈顶指针

C．先后次序无关紧要

D．同时进行

6．在一个栈顶指针为top的链栈中，将一个p指针所指的结点入栈，应执行（ ）。

A．top->next=p;

B．p->next=top->next; top->next=p;

C．p->next=top; top=p;

D．p->next=top->next; top=top->next;

7．在一个栈顶指针为top的链栈中删除一个结点时，用 x保存被删结点的值，则执行（ ）。

A．x=top;top=top->next;

B．x=top->data;

C．top=top->next; x=top->data;

D．x=top->data; top=top->next;

8．一般情况下，将递归算法转换成等价的非递归算法应该设置（ ）。

A．栈 B．队列

C．堆栈或队列 D．数组

9．表达式a\*(b+c)-d的后缀表达式是（ ）。+++

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

10．判断一个顺序队列sq（最多元素为m）为空的条件是（ ）。

A．sq->rear-sq->front== m B．sq->rear-sq->front-1= = m

C．sq->front==sq->rear D．sq->front==sq->rear+1

11．判断一个循环队列Q（最多元素为m）为满的条件是（ ）。

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

C．Q->front==(Q->rear+1)% m D．Q->front!= (Q->rear+1)% m

12．判断栈s满（元素个数最多n个）的条件是（ ）。

A．s->top==0 B．s->top!=0

C．s->top==n-1 D．s->top!=n-1

13．一个队列的入队顺序是a,b,c,d，则离队的顺序是（ ）。

A．a,d,cb B．a,b,c,d C．d,c,b,a D．c,b,d,a

14．如果以链表作为栈的存储结构，则退栈操作时（ ）。

A．必须判断栈是否满 B．判断栈元素类型

C．必须判断栈是否空 D．对栈不作任何判断

15．在解决计算机主机与打印机之间速度不匹配问题时通常设置一个打印数据缓冲区，主机将要输出的数据依次写入缓冲区中，而打印机则从缓冲区中取出数据打印，该缓冲区应该是一个（ ）结构。

A．堆栈 B．队列 C．数组 D．先性表

16．一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

17．从一个栈顶指针为top的链栈中删除一个结点时，用变量x保存被删结点的值，则执行（ ）。

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

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

18．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（ ）。

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

19．在一个链队中，假设f和r分别为队头和队尾指针，则插入s所指结点的运算为（ ）。

A．f->next=s; f=s; B．r->next=s;r=s;

C．s->next=r;r=s; D．s->next=f;f=s;

20．在一个不带头结点的链队中，假设f和r分别为队头和队尾指针，则从该对列中删除一个结点并把结点的值保存在变量x中的运算为（ ）。+++

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

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

21．栈和队列的共同特点是（ ）。

A．都是先进后出 B．元素都可以随机进出

C．只容许在端点处插入和删除元素 D．都是先进先出

22．栈的插入和删除操作在（ ）进行。

A．栈顶 B．栈底

C．任意位置 D．指定位置

23．在一个顺序队列中，队首指针指向队首元素的（ ）位置。

A．前一个 B．后一个

C．当前 D．后面

24．当利用大小为n的数组顺序存储一个队列时，该队列的最大长度为( )。

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

25．从一个顺序存储的循环队列中删除一个元素时，首先需要( )。

A. 队头指针加一 B. 队头指针减一

C. 取出队头指针所指的元素 D. 取出队尾指针所指的元素

1. 判断题

1. 链式栈与顺序栈相比, 一个明显的优点是通常不会出现栈满的情况。（ ）

2. 在一个顺序存储的循环队列中, 队头指针指向队头元素的后一个位置。（ ）

3. 栈和队列都是顺序存取的线性表, 但它们对存取位置的限制不同。（ ）

4. 若让元素1,2,3依次进栈，则出栈次序1,3,2是不可能出现的情况。（ ）

5. 在用单链表表示的链式队列Q中，队头指针为Q->front，队尾指针为Q->rear，则队空条件为Q->front == Q->rear。（ ）

6. 递归定义的数据结构通常用递归算法来实现对它的操作。（ ）

7. 递归的算法简单、易懂、容易编写，而且执行效率也高。（ ）

8. 递归调用算法与相同功能的非递归算法相比，主要问题在于重复计算太多，而且调用本身需要分配额外的空间和传递数据和控制，所以时间与空间开销通常都比较大。（ ）

9. 用非递归方法实现递归算法时一定要使用递归工作栈。（ ）

1. 栈是限定在表的一端进行插入和删除操作的线性表，又称为先进后出表。（ ）
2. 队列的特性是先进后出。（ ）
3. 往栈中插入元素的操作方式是：先写入元素，后移动栈顶指针。（ ）

13.循环队列队头指针在队尾指针前一个位置，队列是“满”状态。（ ）

14. 在队列的顺序存储结构中，当插入一个新的队列元素时，尾指针后移，当删除一个元素队列时，头指针后移。（ ）

1. 向一个栈顶指针为h的链栈(结点的指针域为next)中插入一个s所指结点时，先执行s->next=h，再执行h=s操作。（ ）
2. 一个递归算法不必包括递归终止条件。（ ）
3. 在一个链式队列中，若队头指针与队尾指针的值相同，则表示该队列至多有1个结点。（ ）
4. 在用循环单链表表示的链式队列中，可以不设队头指针，仅在链尾设置队尾指针。（ ）

三、程序选择题

1．在下面空格处填写一条语句，以使下面的链式队列全部元素出队的算法完整。

int write(LinkQueue \*q)

{QueueNode \*p;

if (q->front==q->rear)　 /\*队空\*/

{printf(“队空！无元素可取”);

exit(0);

}

while (q->front->next != NULL)

{p=q->front->next;

q->front->next=p->next; /\*出队\*/

printf(“%4d”,p->data);

free(p); /\*释放已出队结点\*/

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\*队空时，头尾指针指向头结点\*/

}

1. q->front=q->rear;
2. q=q->next;
3. q->rear=q->front;
4. p=p->next;
5. 在下面空格处填写适当的语句，以使下面的循环队列的入队和出队算法完整。

define MAXSIZE 100;

typedef char Elemtype;

typedef struct

{

Elemtype queue [MAXSIZE];

int front,rear;

}sequeuetype;

Sequeuetype Q;

int encqueue(sequeuetype\*Q,elemtype x)

{

if ((Q->rear+1)%MAXSIZE==Q->front)

{

printf(“队列已满!\n”);

return 1;

}

else

{

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

(1)

return 0;

}

} /\*入队算法\*/

Elemtype del\_cqueue(sequeuetype \*Q)

{

if ( (2) )

{

printf(“队列为空!\n”);

return 1;

}

else

{

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

return(Q->queue[Q->front]);

}

} /\*出队算法\*/

1. (1) (Q->rear+1)%MAXSIZE==Q->front (2) Q->front=(Q->front+1)%MAXSIZE;
2. (1) (Q->front+1)%MAXSIZE==Q->rear (2) Q->rear=(Q->rear+1)%MAXSIZE;
3. (1) Q->front==Q->rear (2) Q->queue[Q->rear]=x;
4. (1) Q->queue[Q->rear]=x; (2) Q->front==Q->rear
5. 写出下列程序执行后的结果

SeqQueue Q;

InitQueue(Q);

int a[4]={5,8,12,15};

for(int i=0;i<4;i++) InQueue(Q,a[i]);

InQueue(Q,OutQueue(Q));

InQueue(Q,30);

InQueue(Q,OutQueue(Q)+10);

while(!QueueEmpty(Q)) printf(“%d ”,OutQueue(Q));

执行后的输出结果为：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

1. 5 8 12 15 30
2. 12 15 5 30 18
3. 8 12 15 30 18
4. 12 15 5 18 30
5. 写出下列程序执行后的结果

SeqStack S;

InitStack(S);

Push(S,3);

Push(S,4);

Push(S,5);

int x=Pop(S)+2\*Pop(S);

Push(S,x);

int i,a[4]={5,8,12,15};

for (i=0;i<4;i++) Push(S,a[i]);

while(!StackEmpty(S)) Printf(“%d “,Pop(S));

执行后的输出结果为：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

1. 15 12 8 5 13 3
2. 3 5 8 12 13 15
3. 15 13 12 8 5 3
4. 15 12 13 3 8 5
5. 在下面空格处填写一条语句，以使下面的进栈算法完整。

void Push(struct SeqStack\*s,ElemType x)

{

If (s->top==MaxSize-1){

printf(“栈满溢出错误！\n”);

exit(1);

}

\_\_\_\_\_\_\_\_

s->data[s->top]=x;

}

1. s->top=s->data;
2. s->data++;
3. s->top++;
4. s->data=s->top;
5. 在下面空格处填写一条语句，以使下面的出栈算法完整。

ElemType Pop(struct SeqStack\*s,ElemType x)

{

If (StackEmpty(s)){

printf(“栈下溢出错误！\n”);

exit(1);

}

x=s->data[s->top];

\_\_\_\_\_\_\_\_\_

return x;

}

1. s->top--;
2. s->data--;
3. s->top=s->data;
4. s->data=s->top;

**第四章 字符串**

一、选择题

1．以下陈述中正确的是（ ）。

A．串是一种特殊的线性表 B．串的长度必须大于零

C．串中元素只能是字母 D．空串就是空白串

2．设有两个串p和q，其中q是p的子串，q在p中首次出现的位置的算法称为（ ）。

A．求子串 B．连接

C．匹配 D．求串长

3．串是（ ）。

A．不少于一个字母的序列 B．任意个字母的序列

C．不少于一个字符的序列 D．有限个字符的序列

4．串的长度是指（ ）。

A．串中所含不同字母的个数 B．串中所含字符的个数

C．串中所含不同字符的个数 D．串中所含非空格字符的个数

5. 若串S==“English”，其子串的个数是（ ）。+++

A．9 B．16 C． 36 D．28

6．下面关于串的叙述中，不正确的是（ ）。

A．串是字符的有限序列

B．空串是由空格构成的串

C．模式匹配是串的一种重要运算

D．串即可以采用顺序存储，也可以采用链式存储

7．串与普通的线性表相比较，它的特殊性体现在（ ）。

A．顺序的存储结构 B．链接的存储结构

C．数据元素是一个字符 D．数据元素可以任意

8．空串与空格串（ ）。B

A．相同 B．不相同 C．可能相同 D．无法确定

9．两个字符串相等的条件是（ ）。

A．两串的长度相等

B．两串包含的字符相同

C．两串的长度相等，并且两串包含的字符相同

D．两串的长度相等，并且对应位置上的字符相同

10．串函数Strcat（a,b）的功能是进行串（ ）。

A．比较 B．复制 C．赋值 D．连接

11．串函数StrCmp（“ABCd”，“ABCD”）的值为（ ）。

A．0 B．-1 C．1 D．3

12．设主串为“FABcCDABcdEFaBc”，以下模式串能与主串成功匹配的是（ ）。

A. EFaBc B. ABCdE

C. DABCC D .FAbcC

13．以下四个串中最小的是（ ）。

A．”ABADF” B．”ABAFD”

C．”ABADFA” D．”ABAF”

14．在实际应用中，要输入多个字符串，且长度无法预定。则应该采用（ ）存储比较合适。

A．链式 B． 顺序 C．堆结构 D．无法确定

二、判断题

1. 用字符数组存储长度为n的字符串，数组长度至少为n+1。（ ）

2．串是一种特殊的线性表，其特殊性表现在组成串的数据元素都是字符。（ ）

3．串的两种最基本的存储方式是顺序和链接。（ ）

4．空串的长度是1。（ ）

1. 一个空格的串的长度是0。（ ）

6．两个串相等的充分必要条件是每一个对应位置的字符相同。（ ）

7. 字符串a1=〝heijing〞, a2 =〝hen〞 , a3= 〝heifang〞, a4=“heni〞最小的是a2。（ ）

8．串函数StrCmp（“ABCd”，“ABCD”）的值为-1。（ ）

三、程序选择题

1. 一下程序段的结果是：c的值为（ ）

char \*a[5]={“12378”,”1237”,”1236789”,”1237”,”123708”}

int i,c=0

for(i=0;i<5;i++)

if (strcmp(a[i],”1237”)==0) c++;

A.2 B.5 C.0 D.1237

1. 以下程序段的结果是：c的值为（ ）+++

char a[5]=”1236789”；

int \*p=a,c=0;

While (\*p++) c++;

A.8 B.7 C.10 D.12

**第五章 数组和广义表**

一、选择题

1．一维数组A采用顺序存储结构，每个元素占用6个字节，第6个元素的存储地址为100，则该数组的首地址是（ ）。+++

A．64 B．28

C．70 D．90

2．稀疏矩阵采用压缩存储的目的主要是（ ）。

A．表达变得简单 B．对矩阵元素的存取变得简单

C．去掉矩阵中的多余元素 D．减少不必要的存储空间的开销

3．一个非空广义表的表头（ ）。

A．不可能是原子 B．只能是子表

C．只能是原子 D．可以是子表或原子

4．常对数组进行的两种基本操作是（ ）。

A．建立与删除 B．索引与修改

C．查找和修改 D．查找与索引

5. 设二维数组A[5][6]按行优先顺序存储在内存中，已知A[0][0] 起始地址为1000，每个数组元素占用5个存储单元，则元素A[4][4]的地址为（ ）。

A．1140 B．1145 C． 1120 D．1125

6．设有一个20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a9,2在一维数组B中的下标是（ ）。

A．41 B．32 C．18 D．38

7．广义表的( a , (d,a ,b) , h , (e ,( (i ,j ) ,k )) )深度是（ ）。

A．6 B．10

C．8 D．4

8．广义表( f , h , (a ,b, d, c) , d , e ,( (i ,j ) ,k ) )的长度是（ ）。

A．6 B．10

C．8 D．4

9. 设有一个广义表A (a)，其表尾为（ ）。

A．a B．( （ ） ) C．（） D．(a)

10. 下列广义表中的线性表是（ ）。

A．E(a,(b,c)) B．E(a,E) C．E(a,b) D．E(a,L（ ）)

二、判断题

1. 使用三元组表示稀疏矩阵中的非零元素能节省存储空间。（ ）

2. 一个广义表的表头总是一个广义表。（ ）

1. 一个广义表的表尾总是一个表。（ ）
2. 一个广义表 ( (a), ( (b), c), ( ( (d) ) ) ) 的长度为3，深度为4。（ ）
3. 一个广义表 ( (a), ( (b), c), ( ( (d) ) ) ) 的表尾是 ( (b), c), ( ( (d) ) )。（ ）
4. 需要压缩存储的矩阵可分为特殊矩阵矩阵和稀疏矩阵矩阵两种。（ ）
5. 设广义表L=（（），（）），则其表头是（（））。（ ）
6. 设广义表L=（（），（）），则其表尾是（）。（ ）
7. 设广义表L=（（），（）），则其长度是0。（ ）

10．广义表A（（a,b,c）,(d,e,f)）的表尾为（(d,e,f)）。（ ）

11. 对稀疏矩阵进行压缩存储，矩阵中每个非零元素对应的三元组包括该元素的行号、列号和元素值三项信息。（ ）

12. 设有n阶对称矩阵A，用一维数组s压缩存储A的下三角元素，s的下标从零开始，元素s[26]相应于A中的元素为a7,6。（ ）

**第六章 树和二叉树**

一、选择题

1. 树最适合用来表示（ ）。

A. 有序数据元素　 　　　　　　　　B. 无序数据元素

C. 元素之间具有分支层次关系的数据　 D. 元素之间无联系的数据

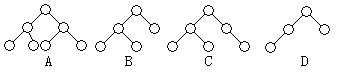
2. 树中所有结点的度等于所有结点数加（ ）。

A. 1 B. 0 C. 2 D. -1

3. 对于一个满二叉树，m个树叶，n个结点，深度为h，则（ ）。

A. n = h + m　 B. h + m = 2n　 C. m = h-1　 D. n = 2 h -1

4. 如图所示一棵二叉树中，（ C ）不是完全二叉树。



5. 深度为5的二叉树至多有（ ）个结点。

A. 16　 B. 32　 C. 31　 D. 10

6. 假定一棵二叉树中，双分支结点数为15，单分支结点数为30，则叶子结点数为（ ）。

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

7. 二叉树第k层上最多有（ ）个结点。

A．2k B．2k-1 C．2k-1 D．2k-1

8.在一棵度具有5层的满二叉树中结点总数为（ ）。

A．31 B．32 C．33 D．16

9. 在一棵二叉树上，第5层的结点数最多为（ ）。

A．8 B．15 C．16 D．32

10.具有127个结点的完全二叉树其深度为（ ）。

A．8 B．7 C．6 D．5

11. 设一棵二叉树中没有度为1的结点，已知叶子结点数为n，此树的结点数为（ ）。

A．2n+2 B．2n+1 C．2n D．2n-1

12.设二叉树中有n2个度为2的结点，n1个度为1的结点，n0个叶子结点，则此二叉树中空指针域个数为（ ）。

A．n0+n1+n2 B．n2+n1+2n0 C．2n2+n1 D．2n0+n1

13.n个结点的二叉树中，用二叉链表做存储，非空指针数目为（ ）。

A．n B．2n C．n-1 D．n+1

14. 在一棵二叉树的二叉链表中，空指针域数等于非空指针域数加（ ）。

A．0 B．1 C．2 D．-1

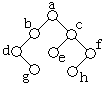
15. 在一非空二叉树的中序遍历序列中，根结点的右边（ ）。

A. 只有右子树上的所有结点　 B. 只有右子树上的部分结点

C. 只有左子树上的所有结点　 D. 只有左子树上的部分结点

16. 如图所示二叉树的中序遍历序列是（ ）。

A. abdgcefh　 B. dgbaechf　 C. gdbehfca　 D. abcdefgh



17. 设n、m为一棵二叉树上的两个结点，中序遍历时n在m前的条件是（ ）。

A. n在m右方　 B. n是m祖先　 C. n在m左方　 D. n是m子孙

18.某二叉树的先序遍历序列和后序遍历序列正好相反，则该二叉树一定（ ）。

A. 空或只有一个结点 B. 完全二叉树

C. 二叉排序树 D. 深度等于其结点数

19.二叉树的先序遍历和中序遍历如下：

先序遍历：EFHIGJK

中序遍历：HFIEJKG

该二叉树根的右子树的根是（ ）。

A．E B．F C．G D．H

20. 如果将给定的一组数据作为叶子数值，所构造出的二叉树的带权路径长度最小，则该树称为（ ）。

A．哈夫曼树 B．平衡二叉树 C．二叉树 D．完全二叉树

21.利用n个值作为叶结点的权生成的哈夫曼树中共包含有（ ）个结点。

A. n B. n+1 C. 2\*n D. 2\*n-1

22. 利用2、4、5、10这四个值作为叶子结点的权，生成一棵哈夫曼树，该树中所有叶子的最长带权路径长度为（ ）。

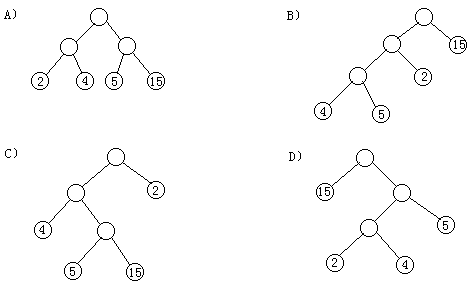
A. 18 B. 16 C. 38 D. 30

23.哈夫曼树是（ ）。

A．满二叉树 B．二叉排序树

C．树的路径长度最短的二叉树 D．带权路径长度最短的二叉树

24.用权值分别为15，2，4，5的四个结点，构造出的哈夫曼树为（ D ）。



25.设哈夫曼树的叶结点数为n，则它的结点总数为（ ）。

A．2n-1 B．2n C．2n+1 D．不确定

二、判断题

1．树是一种线性结构。（ ）

2．树最适合表示元素之间具有层次关系的数据。（ ）

3．如果结点A有 3个兄弟，而且B是A的双亲，则B的度是4。（ ）

4. 树中全部结点的度均大于0。（ ）

5. 森林是m（m≥0）棵互不相交的树的集合。（ ）

6. 深度为k的完全二叉树至少有2k-1个结点。（ ）

7. 完全二叉树中没有度为1的结点。（ ）

8．若树的度为2时，该数为二叉树。（ ）

9．具有三个结点的二叉树有五种。（ ）

10．深度为5的二叉树最多有3层。（ ）

11．具有256个结点的完全二叉树的深度为9  。（ ）

12．具有100个结点的完全二叉树有50 个叶子。（ ）

13. 在二叉树的链接存储中，每个结点设置三个域：值域、左指针域和右指针域。（ ）

14．二叉树只能采用二叉链表来存储。（ ）

15．具有n个结点的二叉树，采用二叉链表存储，共有n+1个空链域。（ ）

16. 二叉树的前序遍历序列中，任意一个结点均处在其子女结点的前面。（ ）

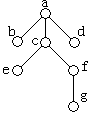
17．已知一棵树的先序序列和后序序列，一定能构造出该树。（ ）

18. 二叉树的遍历就是按照一定次序访问树中所有结点，并且每个结点的值仅被访问一次的过程。（ ）

19. 哈夫曼树一定是完全二叉树或满二叉树。（ ）

20. 哈夫曼树只存在着双支结点，不存在单支结点。（ ）

三、本章综合应用或程序题（都用客观选择形式）

1. 有一棵树如图所示，回答下面问题：

（1）这棵树的根结点是（ ）；

（2）这棵树的叶子结点是（ ）；

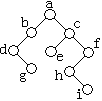
（3）这棵树的度是（ ）；

（4）这棵树的深度是（ ）；

（5）c结点的孩子结点是（ ）；

（6）c结点的父母结点是（ ）。

A. 3 B. 4 C. a D. e、f E. b、e、d、g

2. 由如图所示的二叉树，回答以下问题：（6.4）

（1）其中序遍历序列（ ）；

（2）其前序遍历序列（ ）；

（3）其后序遍历序列（ ）；

A. gdbeihfca B. gbaechif C. abdgcefhi

3. 以3，4，5，8，9，作为叶结点的权，构造一棵哈夫曼树。该树的带权路径长度为（ ）

A. 61 B. 62 C.63 D.65

4. 以2，3，4，7，8，9作为叶结点的权，构造一棵哈夫曼树。权重值为4的叶结点的哈夫曼编码为（ ）

A. 000 B. 001 C.010 D.10

**第七章 图**

一、单选题（共25题）

1. 在一个图G中，所有顶点的度数之和等于所有边数之和的（ ）倍。

A．1/2

B．1

C．2

D．4

1. 一个具有n个顶点的无向完全图包含（ ）条边。

A．n（n−1）

B．n（n+1）

C． n（n−1）/2

D． n（n+1）/2

1. 一个具有n个顶点的有向完全图包含（ ）条边。

A．n（n−1）

B．n（n+1）

C． n（n−1）/2

D． n（n+1）/2

4. 对于一个具有n个顶点和e条边的无向图，若采用邻接表表示，则所有顶点邻接表中的结点总数为（ ）。

A．n

B．e

C．2n

D．2e

5. 在有向图的邻接表中，每个顶点邻接表链接着该顶点所有（ ）邻接点。

A．入边

B．出边

C．入边和出边

D．不是入边也不是出边

1. 邻接表是图的一种（ ）。

A．顺序存储结构

B．链式存储结构

C．索引存储结构

D．散列存储结构

7. 如果从无向图的任一顶点出发进行一次深度优先搜索即可访问所有顶点，

则该图一定是（ ）。

1. 完全图

B．连通图

C．有回路

1. 一棵树

8. 下列有关图遍历的说法不正确的是（ ）。

A．连通图的深度优先搜索是一个递归过程

B．图的广度优先搜索中邻接点的寻找具有“先进先出”的特征

C．非连通图不能用深度优先搜索法

D．图的遍历要求每一顶点仅被访问一次

9. 无向图的邻接矩阵是一个（ ）。

A．对称矩阵

B．零矩阵

C．上三角矩阵

D．对角矩阵

10. 图的深度优先遍历算法类似于二叉树的（ ）遍历。

A．先序

B．中序

C．后序

D．层次

11. G是一个非连通无向图，共28条边，则该图至少有（ ）个顶点。

A. 6

B. 7

C. 8

D. 9

12. 在一个无向图中，若两顶点之间的路径长度为k，则该路径上的顶点数为（ ）。

难度：+

A．k

B．k+1

C．k+2

1. 2k

13. n个顶点的强连通图中至少含有( )。

A．n—l条有向边

B．n条有向边

C．n(n—1)／2条有向边

D．n(n一1)条有向边

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

A．abecdf

B．acfebd

C．aedfcb

D．aebcfd

图1

15. 关键路径是事件结点网络中（ ）。

A. 从源点到汇点的最长路径

B. 从源点到汇点的最短路径

C. 最长的回路

D. 最短的回路

16. 下面（ ）可以判断出一个有向图中是否有环（回路）。

A. 广度优先遍历

B. 拓扑排序

C. 求最短路径

D. 求关键路径

17. 采用邻接表存储的图，其深度优先遍历类似于二叉树的（ ）。

A. 中序遍历

B. 先序遍历

C. 后序遍历

D. 按层次遍历

18. 已知无向图G的顶点数为n，边数为e，其邻接表表示的空间复杂度为（ ）。

A. O(n2)

B. O(n\*e)

C. O(n+e)

D. O(2n)

19. 已知一个图如下图所示，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（ ）。

A. acfdeb

B. acfebd

C. acbdef

D. abecdf



1. 在无向图中定义顶点vi与vj之间的路径为从vi到vj的一个（ ）。

A. 顶点序列

B. 边序列

C. 权值总和

D. 边的条数

21. 在有向图的逆邻接表中，每个顶点邻接表链接着该顶点所有（ ）邻接点。

A. 入边

B. 出边

C. 入边和出边

D. 不是出边也不是入边

22. 设G1=(V1,E1)和G2=(V2,E2)为两个图，如果V1⊆V2,E1⊆E2则称（ ）。

A. G1是G2的子图

B. G2是G1的子图

C. G1是G2的连通分量

D. G2是G1的连通分量

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

A. 将邻接矩阵的第i行删除

B. 将邻接矩阵的第i行元素全部置为0

C. 将邻接矩阵的第i列删除

D. 将邻接矩阵的第i列元素全部置为0

二、判断题

1. 图的深度优先搜索序列和广度优先搜索序列不是惟一的。（ ）

1. 邻接表只能用于存储有向图，而邻接矩阵则可存储有向图和无向图。（ ）

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

4. AOV网是一个带权的有向图。（ ）

5. 从源点到终点的最短路径是唯一的。（ ）

6. 图的生成树是惟一的。（ ）

7. 对连通图进行深度优先遍历可以访问到该图中的所有顶点。（ ）

8. 有n个结点的无向图中，若边数大于n-1，则该图是连通的。（ ）

9. 若一个有向图的邻接矩阵中对角线以下元素均为零，则该图的拓扑有序序列必定存。 （ ）

10. AOV网拓扑排序的结果是惟一的。（ ）

11. 图的广度优先搜索序列是惟一的。（ ）

12. 具有n个顶点的无向图采用邻接矩阵表示，图中的边数等于邻接矩阵中非零元素之和的一半。（ ）

13. 若连通图上各边权值均不相同，则该图的最小生成树是惟一的。（ ）

14. 无向图的邻接矩阵一定是对称的。（ ）

15. 有向图的邻接矩阵一定是非对称的。（ ）

16. 用邻接矩阵存储图的时候，占用空间大小不但与图的结点个数有关还与图的边数有关。（ ）

17. 图的连通分量是无向图的极小连通子图。（ ）

18. 图的强连通分量是无向图的极大连通子图。（ ）

19. 对任意一个图从它的某个顶点出发进行一次深度优先或广度优先搜索遍历可访问到该图的每个顶点。（ ）

1. 一个有向图的邻接表和逆邻接表中的节点个数一定相等。（ ）
2. 有向图用邻接矩阵表示后，顶点i的出度等于第i行中非0且非无穷的元素个数。（ ）

22. 图G的某一最小生成树的代价一定小于其他生成树的代价。（ ）

23. 任一个有向图的拓扑序列只有一个。（ ）

24. 一个无向连通图的生成树是含有该连通图的全部顶点的极小连通子图。（ ）

25. 采用邻接表存储的图的广度优先遍历算法类似于二叉树的按层次遍历。（ ）

三、程序填空题（2题，附加题）

1.已知图G的邻接矩阵如下所示：

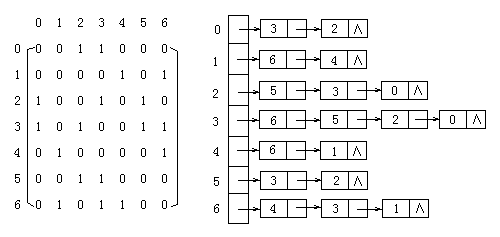
从顶点1出发的广度优先搜索序列为（ ）。

1. 1；2，3， 4；5；6
2. 2；1，3，5；4；6
3. 3；1，2，4；5；6
4. 4；2，3，6；1；5



2.对于一个无向图，假定采用邻接矩阵表示，试分别写出从顶点0出发按深度优先搜索遍历得到的顶点序列。

注：每一种序列都是唯一的，因为都是在存储结构上得到的。



1. 0，2，3，4，5，1，6
2. 0，2，3，5，1，6，4
3. 0，2，3，5，6，1，4
4. 0，2，3，4，5，1，6

**第八章 查找**

一、选择题

1.顺序查找方法适合于存储结构为（ ）的线性表。

A．散列存储 B．索引存储

C．散列存储或索引存储 D．顺序存储或链接存储

2．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。

A．3 B．4 C．6 D．8

3．对二叉排序树进行（ ）遍历，可以使遍历所得到的序列是有序序列。

A．按层次 B．后序 C．中序 D．前序

4．有一个长度为12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．37/12 B．39/12 C．41/12 D．35/12

5．已知一个有序表为{11,22,33,44,55,66,77,88,99}，则顺序查找元素55需要比较（ ）次。

A．3 B．4 C．5 D．6

6.顺序查找法与折半查找法对存储结构的要求是（ ）。

A．顺序查找与折半查找均只适用于顺序表

B．顺序查找与折半查找均既适用于顺序表，也适用于链表

C．顺序查找只是适用于顺序表

D．折半查找适用于顺序表

7．有数据{53,30,37,12,45,24,96}，从空二叉树开始逐个插入数据来形成二叉排序树，若希望高度最小，应该选择的序列是（ ）。

A．45,24,53,12,37,96,30 B．37,24,12,30,53,45,96

C．12,24,30,37,45,53,96 D．30,24,12,37,45,96,53

8.采用顺序查找方法查找长度为n的线性表时，每个元素的平均查找长度为（ ）。

A．n B．n/2

C．(n+1)/2 D．(n-1)/2

9.对于一个线性表，若要求既能进行较快地插入和删除，又要求存储结构能够反映数据元素之间的逻辑关系，则应该（ ）。+++++

A．以顺序存储方式 B．以链接存储方式

C．以索引存储方式 D．以散列存储方式

10.哈希函数有一个共同的性质，即函数值应当以（ ）取其值域的每个值。

A．最大概率 B．最小概率

C．平均概率 D．同等概率

11.在最坏情况下，折半查找与二叉排序树查找性能比较，（ ）

A. 完全相同 B.折半查找性能好

C. 二叉排序树查找性能好 D.不能确定

12.设哈希表长m=14，哈希函数H(key)=key mod 11。表中已有4个结点：addr(15)=4; addr(38)=5; addr(61)=6; addr(84)=7。如用线性探测处理冲突，关键字为49的结点的地址是（ ）。

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

13.采用折半查找方法查找长度为n的线性表时，每个元素的平均查找长度为（ ）。+++++++

A．n2 B.nlog2n C.n D.log2n

14. 哈希表的平均查找长度（ ）

A．与处理冲突的方法有关，与表的长度无关

B．与处理冲突的方法无关，与表的长度有关

C．与处理冲突的方法有关，与表的长度有关

D．与处理冲突的方法无关，与表的长度无关

15.一组记录的关键字是{19,14,23,1,68,20,84,27,55,11,10,79}，用链接地址法构造散列表，散列函数为H(key)=key mod 13,散列地址为1的链中有（ ）个记录。

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

16.二叉排序树的查找效率与二叉树的（ ）有关。

A.高度 B.结点个数 C.树形 D.结点的位置

二、判断题

1．在顺序查找、折半查找、哈希表查找3种方法中，平均查找长度与结点个数n无关的查找方法是折半查找。（ ）

2．在一个查找表中，能够唯一地确定一个记录的关键字称为主关键字。（ ）

3.顺序查找是一种最简单的查找方法。（ ）

4.折半查找的前提条件是，查找表中记录相应的关键字值必须有序或者部分有序。（ ）

5.二叉排序树的建立过程上实际上是从空树逐次插入的过程。（ ）

6.理想情况下，哈希表查找等概率查找成功的时间复杂度是O(1)。（ ）

7.按照一定规则，在二叉排序树上插入、删除结点，仍能保持二叉排序树的性质。（ ）

8.分块查找分为两个步骤：第一步是要对索引表进行查找；第二步是在块中查找。这两步查找都可以采用折半查找或者顺序查找方法。（ ）

9.一个好的哈希函数，应该使哈希地址均匀地分布在整个哈希表的地址区间中，完全避免冲突的发生。（ ）

10.在有序顺序存储的线性表中查找一个元素，用折半查找速度一定比顺序查找快。（ ）

11.二叉树为二叉排序树的充分必要条件是，任一个分支结点的值都大于其左孩子的值，小于右孩子的值。（ ）

12.二叉排序树在呈单支二叉树时，查找效率最低。（ ）

13.将10个元素散列到10000个单元的哈希表中，仍然可能会产生冲突。（ ）

14.根据无序序列构造二叉排序树的过程，也是对无序序列排序的过程。（ ）

15.折半查找方法运用在升序序列比降序序列效率更高，所以降序序列最好先转换为升序序列。（ ）

**第九章 排序**

一、选择题

1. 从未排序序列中依次取出元素与已经排好序的序列中的元素作比较。将其放入已排序序列的正确的位置上，此方法称为（ ）。

A. 插入排序 B. 交换排序

C. 选择排序 D. 归并排序

1. 依次将每两个相邻的有序表合并成一个有序表的排序方法称为（ ）。

A. 插入排序 B. 交换排序

C. 选择排序 D. 归并排序

1. 当两个元素出现逆序的时候就交换位置，这种排序方法称为（ ）。

A. 插入排序 B. 交换排序

C. 选择排序 D. 归并排序

1. 从未排序序列中挑选元素，并将其放入已排序序列的一端，此方法称为（ ）排序。

A. 插入排序 B. 交换排序

C. 选择排序 D. 归并排序

1. 在下列排序方法中，关键字比较的次数与记录初始排列秩序无关的是（ ）。

A. 冒泡排序 B. 希尔排序

C. 选择排序 D. 插入排序

1. 在待排序元素基本有序的情况下，效率最高的排序方法是（ ）。

A. 插入排序 B. 快速排序

C. 堆排序 D. 归并排序

1. 在下列几种排序方法中，平均情况下占用内存量最大的是（ ）方法。

A. 插入排序 B. 选择排序

C. 快速排序 D. 归并排序

1. 每次把待排序的区间划分为左、右两个子区间，其中左区间中记录的关键字均小于等于基准记录的关键字，右区间中记录的关键字均大于等于基准记录的关键字，这种排序称为（ ）。

A. 插入排序 B. 快速排序

C. 堆排序 D. 归并排序

1. 设已有m个元素有序，在未排好序的序列中挑选第m+1个元素，并且只经过一次元素的交换就使第m+1个元素排序到位，该方法是（ ）。

A. 折半插入排序 B. 冒泡排序

C. 归并排序 D. 直接选择排序

1. 对具有n个元素的任意序列采用插入排序法进行排序，排序趟数为（ ）。

A. n-1 B. n

C. n+1 D. [log2n]

1. 对序列（49，38，65，97，76，13，47，50）采用直接插入排序法进行排序，要把第七个元素47插入到已排序中，为寻找插入的合适位置需要进行（ ）次元素间的比较。

A. 3 B. 4

C. 5 D. 6

1. 18个元素进行冒泡法排序，通常需要进行17趟冒泡 ,其中第10趟冒泡共需要进行（ ）次元素间的比较。

A. 7 B. 8

C. 9 D. 10

1. 已知10个数据元素为（54，28，16，34，73，62，95，60，26，43），对该数列从小到大排序，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. 16，28，34，54，62，60，73，26，43，95

D. 28，16，34，54，62，60，73，26，43，95

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

B. 40，38，46，84，56，79

C. 40，38，46，56，79，84

D. 40，38，46，79，56，84

1. 一组记录的关键字序列为（46，20，30，79，56，38，40，84，90，110），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. 40，20，30，38，46，56，79，84，90，110

B. 30，20，40，38，46，84，56，79，90，100

C. 20，30，38，40，46，56，79，84，90，100

D. 20，30，40，38，46，79，56，84，90，100

1. 一组记录的关键字序列为（26，59，36，18，20，25），利用堆排序的方法建立的初始小根堆为（ ）。

A. 18，20，36，59，26，25

B. 18，20，25，59，26，36

C. 26，18，59，20，36，25

D. 26，59，36，18，20，25

1. 一组记录的关键字序列为（80，57，41，39，46，47），利用堆排序（堆顶元素是最小元素）的方法建立的初始堆为（ ）。

A. 39，47，46，80，41，57

B. 41，39，46，47，57，80

C. 39，46，41，57，80，47

D. 39，80，46，47，41，57

1. 一组记录的关键字序列为（46，79，56，38，40，84），利用堆排序的方法建立的初始堆为（ ）。

A. 79，46，56，38，40，84

B. 84，79，56，38，40，46

C. 84，79，56，46，40，38

D. 84，56，79，40，46，38

1. 一组记录的关键字序列为（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. 16，25，48，35，79，82，23，36，40，72

B. 16，25，35，48，23，40，79，82，36，72

C. 16，25，35，48，79，82，23，36，40，72

D. 16，25，35，48，79，23，36，40，82，72

1. 一组记录的关键字序列为（60，47，80，57，39，41，46，30），利用归并排序的方法,对该序列进行(1,1) 归并，即第一趟归并后的结果为（ ）。

A. 47，57，60，80，30，39，41，46

B. 30，39，41，46，47，57，60，80

C. 30，47，80，57，39，41，46，60

D. 47，60，57，80，39，41，30，46

二、判断题

1. n个元素进行冒泡法排序，通常需要进行n-1趟冒泡。（ ）
2. 对16个元素的序列用冒泡排法进行排序，通常需要进行15趟冒泡。（ ）
3. n个元素进行冒泡法排序，通常第j趟冒泡要进行n-j次元素间的比较。（ ）
4. 18个元素进行冒泡法排序，通常需要进行17趟冒泡 ,其中第10趟冒泡共需要进行8次元素间的比较。（ ）
5. 序列3,1,7,18,6,9,13,12经一趟归并排序的结果为1,3,7,18,6,9,13,12。（ ）
6. 待排序的序列为8,3,4,1,2,5,9，采用直接选择排序算法，当进行了两趟选择后，结果序列为1,2,8,3,4,5,9。（ ）
7. 序列15,13,16,14,19,17，采用冒泡排序算法(升序)，经一趟冒泡后，结果序列是13,15,14,16,17,19。（ ）
8. 在对10个记录的序列(14,30,10,7,22,13,66,85,47,58)进行直接插入排序时，当把第6个记录13 插入到有序表时，为寻找插入位置，需比较3次。（ ）
9. 冒泡排序是一种比较简单的插入排序方法。（ ）
10. 在归并排序中，在第3趟归并中，是把长度为4的有序表归并为长度为8的有序表。（ ）

三、综合应用选择题空题（把合适的选项编号填写在括号内）

1.

（1）一组记录的关键字序列为（45，40，65，43，35，95），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为（ ）。

A. 35，40，65，45，35，95

B. 35，40，65，43，45，95

C. 35，40，43，45，65，95

D. 35，40，45，43，65，95

（2）对上述序列利用直接插入排序，逐次插入过程中，共进行了（ ）次元素间的比较。

A. 8 B. 11      C. 9      D. 10

2.

（1）对关键字序列（36，69，46，28，30，74）采用快速排序，以第一个关键字为分割元素，经过一次划分后的结果序列为（ ）。

A. 30，28，46，36，69，74       B. 28，30，36，46，69，74

C. 28，30，46，36，69，74          D. 30，28，36，46，69，74

（2）用冒泡法对上述序列排序，经过两趟冒泡的结果序列为（ ）。

    A. 36，28，30，46，69，74            B. 36，46，28，20，69，74

    C. 38，36，30，46，69，74            D. 28，36，30，46，69，74

3.

（1）一组记录的关键字序列为（47，80，57，39，41，46），利用堆排序的方法建立的初始堆为（ ）（堆顶元素是最小元素,采用树的形式建堆）。

  A. 39，41，57，80，47，46        B. 39，41，46，80，47，57

C. 39，47，46，80，41，57        D. 39，41，57，80，46，47

（2）输出堆顶元素后,调整后的堆为（ ）。

A. 41，47，46，80，57          B. 41，57，46，80，47

C. 41，57，80，47，46          D. 41，80，46，47，57

4.

（1）一组记录的关键字序列为（42，37，62，40，32，92），利用快速排序算法，以第一个关键字为分割元素，经过一次划分后结果为（ ）。

A. 37，32，40，42，62，92          B. 32，37，40，42，62，92

C. 32，37，40，62，42，92               D. 42，37，40，62，32，92

（2）利用筛选过程把序列（42，82，67，102，16，32，57，52）建成初始堆（小根堆）为（ ）。

A. 42，16，67，52，82，32，57，102

B. 16，32，42，52，82，57，67，102

C. 16，42，32，52，82，67，57，102

D. 16，32，82，52，42，102，67，57

5.

（1）对关键字序列（56，51，71，54，46，106），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. 46，51，56，54，71，106          B. 56，51，54，46，71，106

C. 46，51，54，56，71，106               D. 56，51，46，54，71，106

（2）一组记录的关键字序列为（60，47，80，57，39，41，46，30），利用归并排序的方法，经过(2,2)归并的结果序列为（ ）。

A. 30，57，60，80，47，39，41，46

B. 47，,60，57，80，30，39，41，46

C. 41，57，60，80，30，39，47，46

D. 47，57，60，80，30，39，41，46

四、程序选择题空题（从答案选项中选择合适的内容，完成程序中的空格）

1. 以下直接插入排序算法对存放在a[0],a[1],···,a[n-1]中，长度为n的记录序列按关键字key由小到大排序。

void disort (NODE a[ ], int n)

{ int i,j;

NODE temp;

for (i=1;i<n;i++)

{ temp=a[i];

j=j-1;

while (\_\_(1)\_\_\_\_&&temp.key<a[j].key)

{ a[j+1]= (2) ;

(3) ;

}

a[j+1]=  \_\_(4) ;

}

}

【答案选项】

A. j--

B. j>=0

C. temp

D. a[j]

2. 以下程序是折半插入排序的算法

    设待排序的记录序列存放在a[1],…a[n]中，以a[0]作为辅助工作单元，程序是要把a[i] 插入到已经有序的序列a[1],…a[i-1]中。

   void binsort (NODE a[ ],int n)

   {   int x,i,j,s,k,m;

       for (i=2；i<=\_\_(1)\_\_\_\_ ;i++)

       {  a[0]=a[i];

           x= a[i].key;

           s=1;

           j=i-1;

           while (s<=j)

           {  m=\_\_(2)\_\_\_;

              if( x<a[m].key)

                \_\_(3)\_\_\_;

              else

                 \_\_(4)\_\_\_;

            }

           for ( k=i-1;k>=j+1;k- -)

             \_\_(5)\_\_\_=a[k];

         a[j+1]=a[0];

       }

    }

【答案选项】

A. (s+j)/2

B. j=m-1

C. a[k+1]

D. n

E. s=m+1

3. 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，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) 。

【答案选项】

A. a[i]=a[i+1]

B. j<=n-1

C. a[i+1]=temp

D. 当某趟冒泡中没有出现交换则已排好序结束循环

E. i<=n-j

4. 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。

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)\_\_ \_\_;

}

}

}

【答案选项】

A. n

B. a[i]=a[k]

C. k=j

D. a[k]=temp

E. n-1

5. 以下程序是快速排序的算法

设待排序的记录序列存放在a[start],…a[end]中，按记录的关键字进行快速排序，先进行一次划分，再分别进行递归调用。

void quicksort ( NODE a[ ], int start ,int end )

{ int i，j;

NODE mid ;

if (start>=end )

return;

i=start;

j=end;

mid=a[i];

while (i<j)

{ while(i<j && a[j].key>mid.key)

j- -;

if(i<j)

{ a[i]=a[j];

\_\_\_(1)\_\_\_\_\_;

}

while(i<j && a[i].key<=mid.key)

\_\_\_(2)\_\_\_\_\_;

if(i<j)

{ \_\_\_(3)\_\_\_\_\_;

\_\_\_(4)\_\_\_\_\_;

}

}

a[i]=mid;

quicksort (a,stat, i-1);

quicksort \_\_\_(5)\_\_\_\_\_;

}

【答案选项】

A. a[j]=a[i]

B. (a, i+1,end)

C. i++

D. j--