**第5章 内部分类 习题参考答案**

**四、习题**

**一、填空题**

1. 大多数排序算法都有两个基本的操作： 比较 和移动。
2. 评价排序算法优劣的主要标准是 时间复杂度 和算法所需的附加空间。
3. 根据被处理的数据在计算机中使用不同的存储设备，排序可分为： 内排序 和外排序。
4. 外排序是指在排序过程中，数据的主要部分存放在计算机的 外存 中。
5. 对n个关键字进行冒泡排序，其可能的最小比较次数为： n-1 次。
6. 在最坏情况下，在第i趟直接插入排序中，要进行 i-1 次关键字的比较。
7. 对n个关键字进行冒泡排序，时间复杂度为 O(n2) 。
8. 快速排序在最坏情况下的时间复杂度是 O(n2) 。
9. 对于n个记录的集合进行归并排序，所需要的平均时间为： O(log2n) 。
10. 对于n个记录的集合进行归并排序，所需要的附加空间是 O(n) 。
11. 若原始数据接近无序，则选用 快速排序 最好。
12. 在排序前，关键字值相等的不同记录，排序后相对位置保持 不变 的排序方法，称为稳定排序方法。
13. 在插入排序和选择排序中，若初始数据基本正序，则选用 插入排序 较好。
14. 当增量为1时，该趟希尔排序与 直接插入 排序基本一致。
15. 第一趟排序后，序列中键值最大的记录交换到最后的排序算法是 冒泡排序 。
16. 依次将每个记录插入到一个有序的子文件中的排序方法称为 直接插入 排序。
17. 在插入排序、选择排序和归并排序中，排序是不稳定的为： 选择排序 。
18. 在对一组记录（54，38，96，23，15，72，60，45，83）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 3 次。
19. 两个序列分别为：

L1={25，57，48，37，92，86，12，33}

L2={25，37，33，12，48，57，86，92}。

用冒泡排序法对L1和L2进行排序，交换次数较少的是序列： L2 。

（20）对一组记录（54，35，96，21，12，72，60，44，80）进行直接选择排序时，第四次选择和交换后，未排序记录是 54，72，60，96，80 。

**二、选择题**

（1）排序是根据（ A ）的大小重新安排各元素的顺序。

A．关键字 B．数组 C．元素件 D．结点

（2）评价排序算法好坏的标准主要是（ D ）。

A．执行时间 B．辅助空间

C．算法本身的复杂度 D．执行时间和所需的辅助空间

（3）直接插入排序的方法是（ B ）的排序方法。

A．不稳定 B．稳定 C．外部 D．选择

（4）直接插入排序的方法要求被排序的数据（ B ）存储。

A．必须链表 B．必须顺序 C．顺序或链表 D．可以任意

（5）排序方法中，从无序序列中选择关键字最小的记录，将其与无序区（初始为空）的第一个记录交换的排序方法，称为 ( D )。

A．希尔排序 B．归并排序 C．插入排序 D. 选择排序

（6）每次把待排序方的区间划分为左、右两个区间，其中左区间中元素的值不大于基准元素的值，右区间中元素的值不小于基准元素的值，此种排序方法叫做（ C ）。

A．冒泡排序 B．堆排序 C．快速排序 D. 归并排序

（7）快速排序在（ C ）情况下最易发挥其长处。

A．待排序的数据中含有多个相同的关键字 B．待排序的数据已基本有序

C．待排序的数据完全无序 D．待排序的数据中最大值与最小值相差悬殊

（8）下述几种排序方法中，要求内存量最大的是：（ D ）。

A．插入排序 B．选择排序 C．快速排序 D. 归并排序

（9）直接插入排序的方法是从第（ B ）个元素开始，插入到前边适当位置的排序方法。

A．1 B．2 C．3 D．n

（10）堆的形状是一棵（ C ）。

A．二叉排序树 B．满二叉树 C．完全二叉树 D．平衡二叉树

（11）内排序是指在排序的整个过程中，全部数据都在计算机的（ A ）中完成的排序。

A．内存 B．外存 C．内存和外存 D．寄存器

（12）快速排序的方法是（ A ）的排序方法。

A．不稳定 B．稳定 C．外部 D．选择

（13）下列排序方法中，关键字比较次数与记录的初始排列次序无关的是（ A ）。

A．选择排序 B．希尔排序 C．插入排序 D．冒泡排序

（14）下述几种排序方法中，平均时间复杂度最小的是（ A ）。

A．希尔排序 B．插入排序 C．冒泡排序 D．选择排序

（15）对有n个记录的表作快速排序，在最坏情况下，算法的时间复杂度是（ B ）。

A．O(n) B．O(n2) C．O(nlog2n) D．O(n3)

（16）冒泡排序的方法对n个数据进行排序，第一趟排序共需要比较（ C ）次。

A．1 B．2 C．n-1 D．n

（17）对ｎ个不同的排序码进行冒泡（递增）排序，在下列（ B ）情况比较的次数最多。

A．从小到大排列好的 B．从大到小排列好的 C元素无序 D．元素基本有序

（18）用直接插入排序法对下面的四个序列进行由小到大的排序，元素比较次数最少的是（ B ）。

A，94，32，40，90，80，46，21，69 B．21，32，46，40，80，69，90，94

C．32，40，21，46，69，94，90，80 D．90，69，80，46，21，32，94，40

（19）一组记录的排序码为(25，48，16，35，79，82，23，40)，其中含有4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 ）。

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

（20）一个数据序列的关键字为：（46，79，56，38，40，84），采用快速排序，并以第一个数为基准得到第一次划分的结果为：（ C ）

A．（38，40，46，56，79，84） B．（40，38，46，79，56，84）

C．（40，38，46，56，79，84） D．（40，38，46，79，56，84）

**三．排序过程分析**

（1）已知数据序列{10，8，18，15，7，16}，写出采用直接插入算法排序时，每一趟排序的结果。

解： 10 8 18 15 7 16

第一趟结束时结果： [8 10] 18 15 7 16

第二趟结束时结果： [8 10 18] 15 7 16

第三趟结束时结果： [8 10 15 18] 7 16

第四趟结束时结果： [7 8 10 15 18] 16

第五趟结束时结果： [7 8 10 15 16 18]

（2）已知数据序列{18，17，60，40，07，32，73，65}，写出采用直接插入算法排序时，每一趟排序的结果。

解： 18 17 60 40 07 32 73 65

第一趟结束时结果： [17 18] 60 40 07 32 73 65

第二趟结束时结果： [17 18 60] 40 07 32 73 65

第三趟结束时结果： [17 18 40 60] 07 32 73 65

第四趟结束时结果： [07 17 18 40 60] 32 73 65

第五趟结束时结果： [07 17 18 32 40 60] 73 65

第六趟结束时结果： [07 17 18 32 40 60 73] 65

第七趟结束时结果： [07 17 18 32 40 60 65 73]

（3）已知数据序列{17，18，60，40，7，32，73，65，85}

请写出采用冒泡排序法对该序列作升序排序时每一趟的结果。

解： 17 18 60 40 7 32 73 65 85

第一趟排序结果： 17 18 40 7 32 60 65 73 85

第二趟排序结果： 17 18 7 32 40 60 65 73

第三趟排序结果： 17 7 18 32 40 60 65

第四趟排序结果： 7 17 18 32 40 60

第五趟排序结果： 7 17 18 32 40

第五趟排序过程中已无记录交换，排序结束。

（4）已知数据序列{80，18，9，90，27，75，42，69，34}

请写出采用冒泡排序法对该序列作升序排序时每一趟的结果。

解： 80 18 09 90 27 75 42 69 34

第一趟排序结果： 18 09 80 27 75 42 69 34 90

第二趟排序结果： 09 18 27 75 42 69 34 80

第三趟排序结果： 09 18 27 42 69 34 75

第四趟排序结果： 09 18 27 42 34 69

第五趟排序结果： 09 18 27 34 40

第六趟排序结果： 09 18 27 34

第六趟排序过程中已无记录交换，排序结束。

（5）已知数据序列{10，18，4，3，6，12，9，15，8}，写出希尔排序每一趟排序的结果。（设d=5、2、1）

解：

10 18 4 3 6 12 9 15 8

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

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

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

（6）已知数据序列{12，02，16，30，28，10，17，20，06，18}，写出希尔排序每一趟排序的结果。（设d=5、2、1）

解： 12 02 16 30 28 10 17 20 06 18

d=5

10 02 16 06 18 12 17 20 30 28

d=2

12 02 16 06 17 12 18 20 30 28

d=1 02 06 10 12 16 17 18 20 28 30

（7）已知数据序列{10，18，4，3，6，12，9，15}，写出二路归并排序的每一趟排序结果。

[10] [18] [4] [3] [6] [12] [9] [15]

[10 18] [3 4] [6 12] [9 15] 第一趟排序结果

[3 4 10 18] [6 9 12 15] 第二趟排序结果

[3 4 6 9 10 12 15 18] 第三趟排序结果

（8）已知数据序列{53，36，48，36，60，7，18，41}，写出采用简单选择排序的每一趟排序结果。

解：[53 36 48 36 60 7 18 41]

（7） [36 48 36 60 53 18 41]

（7 18） [48 36 60 53 36 41]

（7 18 36） [48 60 53 36 41]

（7 18 36 36） [60 53 48 41]

（7 18 36 36 41） [53 48 60]

（7 18 36 36 41 48） [53 60]

（7 18 36 36 41 48 53） [60]

（7 18 36 36 41 48 53 60 ）

（9）已知数据序列{10，1，15，18，7，15}，试画出采用快速排序法，第一趟排序的结果。

解：

10 1 15 18 7 15

low high

交换

7 1 15 18 [10] 15

low high

交换

第一趟排序结果： 7 1 [10] 18 15 15

low high

（10）已知数据序列{10，1，15，18，7，15}，试写出采用快速排序法，第一趟排序的结果。

解：

7 1 10 18 15 15

**四、程序填空**

void BInsSort( ) //按递增序对R[1]~R[ n ]进行二分插入排序

{ int i, j, low, high, m;

for( i=2;i<= n ; i++)

{ R[0]=R[i]; // 设定R[0]为监视哨

low=1;

high= n ；

while(low <= high)

{ m=(low+high)/2 ;

if(R[0]<R[m])

high=m-1 ;

else

low=m+1;

}

for(j=i-1;j>=high+1;j--)

R[j+1]= R[ j ] ; // 元素后移

R[high]=R[0]; // 插入

}

}

**五、 算法题**

1．以单链表为存储结构，写一个直接选择排序算法。

解： void selectsort(pointer h)

{ pointer p,q,r,s,t;

t=NULL;

while(h)

{ p=h;q=NULL;

s=h;r=NULL;

while(p)

{ if(p->key<s->key)

{ s=p;p=q; }

if(s==h)

h=h->link;

else

h=s;

s->link=t;

t=s;

}

h=t;

}

}

2．以单链表作为存储结构实现直接插入排序算法。

解： void InsertList(List head)

{ Lnode \*p, \* pprev**,**q,\*qprev, \*current;

if(!head)

return;

pprev=head;

p=head->next;

while (p)

{ q=head;

qprev=NULL;

while(q->key <p->key) // 查找插入位置

{qprev=q;q=q->next;}

if(q= =p) // p最大，无须插入

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

else

{ current=p; p=p->next;

pprev->next=p;

current->next=q;

if(q==head) // 插在表头

head=current;

else // 插在中间某个位置上

qprev->next=current;

}

}

}

3．设计一个算法，使得在尽可能少的时间内重排数组，将所有取负值的关键字放在所有取非负值的关键字之前。

解： void part (int a[ ])

{ i=1;j=n; // 初、终下标

while(i<j)

{ while(i<j && a[j]>=0) // 自右向左找非负数

j--;

while(i<j && a[i]<0) // 自左向右找负数

i++;

if(i<j)

{ t=a[i];

a[i]=a[j];

a[j]=t;

i++;

j--;

}

}

}

4．

设已排序的文件用单链表表示，再插入一个新记录，仍然按关键字从小到大的次序排序，试写出该算法。

void insert(lklist head;datatype x)

{

s=new( node );

s->key=x;

s->next= NULL;

if(head= =NULL)

head=s;

else

{ p=head;

q= NULL;

while(( p! =NULL) && (s->key > p->key ))

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

if(q= =NULL)

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

else

{ if(p==NULL)

q->next=s;

else

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

}

}

}