北京大学成人教育理科类计算机教材

语言应用教程

吕凤翥 吕波 编著

C语言应用教程

北京大学
出版社

北京大学成人教育
理科类计算机教材

C 语言应用教程

吕凤翥 吕 波 编著

北京大学出版社
北 京

内 容 简 介

本书是北京大学成人教育学院理科类各专业教材，根据教学大纲的要求，由主讲教师精心编写。读者在初步具备C语言基础知识后，通过本书能进一步学习和掌握C语言编程的方法和技巧，从而提高对C语言的应用水平和编程能力。

本书包括了C语言程序设计在数据结构、计算绘图、窗口菜单设计和统计计算等方面的应用。本书在各种应用中，从讲清基本概念入手，介绍常用的算法，并列举大量的程序实例。通过分析和研究实例，可以学到C语言编程的方法和技巧。许多实例可以直接用于你正在从事的或将来准备从事的实际应用项目中去。本书每章后面提供了一定量的练习题和作业题，认真做好可以帮助掌握本书的内容。

本书适用于大专院校，成人教育，自学考试和等级考试作为教材，也适用于从事计算机工作的技术人员作为自学参考书。

图书在版编目(CIP)数据

C语言应用教程/吕凤翥等编著. 北京：北京大学出版社，1997.5
ISBN 7-301-03443-1
Ⅰ. C… Ⅱ. 吕… Ⅲ. C语言-应用-成人教育：高等教育-教材 Ⅳ. TP311.2

JS573/10

书　　名：C语言应用教程
著作责任者：吕凤翥　吕波
责 任 编 辑：杨锡林
标 准 书 号：ISBN 7-301-03443-1/TP.0340
出　版　者：北京大学出版社
地　　址：北京市海淀区中关村北京大学内　100871
电　　话：出版部 62752015　发行部 62559712　编辑部 62752032
排　版　者：北京盛达激光照排中心
印　刷　者：北京经纬印刷厂印刷
发　行　者：北京大学出版社
经　销　者：新华书店
787×1029 毫米　16 开本 20.375　印张 510　千字
1997 年 5 月第一版　1997 年 5 月第一次印刷
定　　价：28.00 元

前　言

C 语言在计算机编程应用中越来越显示出重要的作用，它是一种应用广泛实用性强的高级语言。因此，C 语言越来越受到广大编程者的厚爱。四年前，作者在北京大学出版社曾出版过一本《实用 C 语言基础教程》，它着重讲述了 C 语言的基础知识和基本编程方法，深受读者欢迎。今天，又将近些年来从事 C 语言教学和科研中积累的成果及经验，编写成书献给读者。谨此为推广和普及 C 语言在实际中的应用做出一点贡献。

本书共分五章，前四章分别介绍 C 语言在不同领域中的具体应用。第一章讲述 C 语言在数据结构方面的应用，详细讲述了如何使用 C 语言编写排序和检索程序，以及用 C 语言编写队列、栈、链表和树等复杂数据结构的应用程序。第二章讲述 C 语言编程在计算机绘图方面的应用，列举了 Turbo C 2.0 编译系统所提供的屏幕函数和图形函数的书写格式和使用方法，并使用这些函数绘制出丰富多采的图表和画面，同时还介绍了利用 PC-DOS 操作系统中的 ROM-BIOS 资源和直接存取视频缓冲区方法编写绘图函数的方法。第三章讲述了 C 语言编程在窗口和屏幕菜单设计方面的应用，通过具体实例讲述了弹出窗口和屏幕菜单设计的编写方法，实用性较强。第四章讲述了 C 语言编程在统计计算方面的应用，书中列举了 C 语言在基本统计量、统计简图、回归和预测以及随机数等方面的编程实例。通过这些实例展示出 C 语言在统计计算方面的图文并茂的效果。本书第五章总结了 C 语言编程的特点及风格，C 语言程序的优化方法和调试 C 语言的方法以及在 C 语言编程和调试中常见的错误及纠正方法，该章最后通过一个较大的应用程序作为结束，从而反映出 C 语言编程的方法。

本书的第二、三、五章由吕凤翥编写，第一、四章由吕波编写。全书由吕凤翥负责统一和审编。

由于本书涉及到 C 语言在各个领域的应用，为使广大读者能够接受所讲的内容，尽量简化专业性强的概念，而采用通俗易懂的语言来描述这些概念，并选用了具有 C 语言特点的算法，通过大量例子加以说明，使读者从中较快地学到编程方法和技巧，并可运用于解决实际问题。在编写中参阅了一些资料，从中吸取精华，力求简捷，突出重点，注重实用。本书中所有的例题都在 586 微机上 Turbo C 2.0 版本 C 语编译系统下调试通过。

书中每章后面都有练习题和作业题。练习题是为了读者掌握好本章重点内容而设置的，读者学完某章后，可用本章后的练习题检验一下自己的学习效果。作业题是提供给读者动手编程上机调试用的题目。希望读者能够认真做好练习和作业题，它将有助于掌握本书的内容。另外，本书中所有例题都存放在备好的软盘中。

由于时间较紧，本人实际经验和水平有限，书中难免有错误，欢迎批评指正。

作　者

1996 年 12 月于北京大学燕北园

目　录

第一章　C语言与数据结构

本章主要讲述C语言在数据结构方面的应用。数据结构和算法是当今计算机科学研究的基本课题，两者之间有着十分紧密的联系，一个好的程序离不开好的数据结构和好的算法。在编程中，讲到一种类型的数据结构时，总是离不开对这种类型数据结构所需要进行的运算或操作，也只有通过对这些运算的算法的研究，才可能进一步搞清这种数据结构的作用。反过来，在讲述一种算法时，也总是要联系到该算法所处理的对象和结果的数据结构。因此，学习和掌握数据结构对学会计算机编程具有重要的基础作用。

什么是数据结构呢？一般认为它包含如下三个方面：(1) 数据的逻辑结构，即按某种逻辑关系组织起来一组数据；(2) 数据的存储结构，即按一定的存储方式把一组数据存储在计算机的存储器中；(3) 数据的运算，即在一组数据上定义一个运算的集合。这便是数据结构的概念和含义。

本章中介绍一些常见的线性结构，例如：队列、栈和链表，以及非线性结构，例如：二叉树。排序和检索是数据结构的常见的运算，也是本章讨论的主要内容。

1.1　排序和检索

本节主要讨论排序和检索这两种运算。排序运算是将线性结构中的结点按照某种指定的顺序进行重新排列。检索运算是在数据结构中查找满足一定条件的结点，这里讲到的结点是指用来组成结点的元素。结构一般说来是由若干个结点及其相互间的关系所组成的。一般又把一个结点看成为包含了若干个字段。结点的值及结点之间的关系一起被存放到计算机中，它们是机器处理的对象，是某种类型的数据。

1.1.1　排序和检索的概念

1. 排序的种类和方法

排序在数据结构中是一种重要的运算。排序可以分成两种类型：内排序和外排序。内排序主要是处理内存中的数据；外排序不仅处理内存中数据，还要处理外存中的文件。这里，着重讨论内排序。

由于对排序所采用的策略不同，排序的方法很多。人们经常使用的方法有如下三种：

(1) 交换排序法

交换排序的基本方法是：两两比较待排序记录的排序码，并交换不满足顺序要求的那些数据对，直到全部满足要求为止。

常用的交换排序法有起泡排序法(又称上推排序法)和快速排序法。

① 起泡排序是一种最简单的排序方法。对于$k_1,k_2,\cdots,k_n$的n个数据组成的序列，使用起泡排序的具体做法是：先比较k_1和k_2，如果$k_1>k_2$，则交换k_1和k_2；否则不交换。然后对k_2(可能是刚交换来的)和k_3作同样的比较和处理，重复此过程直到处理完k_{n-1}和k_n。这样从(k_1，

k_2)到(k_{n-1},k_n)的 n－1 次比较和交换的过程称为第一次起泡。这步的作用是将最大的 k_i 传到最后，重复执行这种处理，不过第二次起泡只须从(k_1,k_2)到(k_{n-2},k_{n-1})作 n－2 次比较和交换，于是第二大的又移到了后面，依此类推，最多做 n－1 次起泡便可实现全部排序。

此种排序最大的比较次数为$\frac{1}{2}n(n-1)$，其中，n 是数据项个数；最坏情况下，交换次数为$\frac{3}{2}n(n-1)$；它们都是 n^2 的数量级。

② 快速排序，又称为分区交换排序，它是较快的一种排序。快速排序的基本思想是：在待排序的 n 个数据中，任选一个数据(例如选第一个数据)。以该数据为标准，将所有数据分成两组。第一组中各数据都小于等于该标准数据，第二组中各数据都大于该标准数据，并把该标准数据排在这两组数据的中间，这也是该标准数据最终排放的位置。然后，对这两组数据分别重复上述方法，直到所有数据都排到相应位置为止。

此种排序最大的比较次数约为 $n \cdot \log n$；最大的交换次数约为$\frac{n}{6} \cdot \log n$。它与其他方法比较起来，执行时间较短。

(2) 插入排序法

插入排序的基本方法是：每一步将一个待排序的数据，按其值的大小插入到前面已经排好序的数据序列的适当位置，直到全部插入完为止。这里仅介绍两种常用的插入排序方法：直接插入排序法和 Shell 排序法。

① 直接插入排序首先对数据序列中的前两个数据进行排序，然后根据前两个数据排序情况将第 3 个数据插入到适当位置。第三步再根据前三个数据排序的情况把第 4 个数据插入到适当位置。依次类推，直到把所有数据都排好序为止。

此种排序方法最小比较次数约为 n，最大比较次数约为$\frac{1}{2}n^2$；最小交换次数约为 2n，最大交换次数约为$\frac{1}{2}n^2$。

② Shell 排序法又称为缩小增量法。该方法是由 D. L. Shell 在 1959 年提出来的。它的具体操作方法是：在 n 个数据项组成的数据序列中，首先取一个整数 $d_1<n$，把全部数据项分成 d_1 组，所有相隔距离为 d_1 倍数的数据项放在一组中，在各组内进行排序；然后，取 $d_2<d_1$ 重复上述分组和排序工作；直到 $d_i=1$ 即所有数据放在一组中排序为止。各组内排序可采用直接插入法。一般说来，开始时 d 取值较大，每组数据项数较少，所以排序比较快。后来 d_i 变小，每组数据项增多，但由于已按 d_{i-1}作为距离排好序，所以，后来较易排序。

数据项之间的间隔称为增量。每步所用增量可以改变，唯一的要求是最后一步增量为 1。关于增量 d_i 的选择看法不一，有人认为选取奇数为好，也有人认为之间互素为好。例如，增量序列选取为 9,5,3,1 使用起来效果较好。

此种排序方法平均比较次数和平均交换次数都在 $n^{1.3}$左右，这种方法在执行时间上有了较大改进。

(3) 选择排序法

选择排序的基本方法是：每步从待排序的数据项中选出其值为最小的数据，把它顺序放在已排序的数据序列的最后，直到全部排序好为止。这里仅介绍一种直接选择排序法。

直接选择排序法是一种很简单的排序方法，具体操作方法如下：第一步挑选出最小的数据

项，将其与第一个数据项交换。第二步从剩下的n－1个中选出最小的，将其与第二个数据项交换，依此类推，直到所有数据排好序为止。

此种排序方法要比较$\frac{1}{2}(n^2-n)$次；交换次数最好为3(n－1)次，最坏为$\frac{n^2}{4}+3(n-1)$次。它与起泡排序法比较，比较次数相同，交换次数减少。

2. 排序算法的选择

以上介绍了5种不同的排序方法，每种方法都有其优点和不足。如何判断它们的优劣，又怎样在实际中选择排序方法？这里有下述几点应该考虑：

① 排序的速度。速度快慢至关重要，排序的速度与排序操作中比较次数和交换次数有关，而交换所需的时间更多。

② 排序对象的情况。如果排序对象开始就已有序，则执行起来省力；如果排序对象的有序性逐渐变差，则执行起来费劲。可根据排序对象的自然情况，选择排序方法。

在上述5种排序方法中，目前人们认为最好的方法是快速排序法，其次是Shell排序法。剩下的三种排序法为直接插入排序法、直接选择排序法和起泡排序法。

总之，选择排序算法主要看执行时所需要的时间，其次还要考虑执行时所需附加的空间，在数据量小时，快速排序的递归调用所带来的附加负担会降低其算法的效率。另外，还应考虑算法本身的复杂程度。

3. 检索的种类和方法

检索是数据处理中常用的一种运算。所谓检索就是在数据结构中查找出满足某种条件的结点。例如，在学生成绩表中，查找某个学号的学生的成绩，学号是学生成绩表中的关键码。检索的结果有两种可能，一种是没有查找到满足条件的结点，这时称为“检索失败”；另一种是找到了满足条件的结点，这时称为“检索成功”。

在实际中，对不同的存储结构，有不同的检索方法，因此，在选择检索方法时，应先搞清这种方法所需要的存储结构。常用的检索方法有顺序检索法和折半检索法。这两种方法都是线性表的检索方法。

① 顺序检索法是一种最简单的检索方法，它常用来对于没有排序的数据序列进行检索。其检索方法是从数据序列的第一个元素开始，将数据与要查找的数据进行比较，如果相等(即查找到)或者查找到数据序列末尾时，查找过程停止。

这种检索方法的优点是算法简单，对被检索数据序列没有排序要求。缺点是检索时间长。

② 折半检索法，又称为二分法检索。这是一种效率较高的检索方法。检索时要求被检索的数据序列先按其关键码值的大小排序。这种检索方法的具体操作是：先用要查找的关键码值与数据序列表的中间位置的关键码值相比较，这个中间位置结点将表分成两个子表。比较结果，如果相等，则检索结束；如果不相等，再根据要查找的关键码值比较中间位置结点的关键码值的大小，确定下一步要检索哪个子表，这样递归进行下去直到查找到满足条件的结点，或者确定该表中没有满足条件的结点时为止。

这种检索方法的优点是检索时间短，即效率较高。它的缺点是要求被检索的数据序列先要按其关键码值大小排序，并要求顺序存储。

此外，还有分块检索法、散列表检索法等其他方法，本书就不再介绍。

衡量一种检索算法效率的主要标准是：(1) 平均比较次数，又称为平均检索长度，通常用

E(n)表示，其中n是检索数据表中数据项的个数。该值越小，则执行检索所需时间越短。(2) 所需的存储量，要求空间开销越小越好。(3) 算法的复杂性，希望算法越简单越好。在实际中上述因素往往要综合考虑。

1.1.2 排序的各种方法及其实现

1. 交换排序法

(1) 起泡排序法

假定待排序的数据序列如下所示，按照前面介绍过的操作方法，具体执行过程如图1-1所示。

待排序的数据序列为：

56 82 47 25 93 13 18 62 75 5

该序列共有10个数据项。

序号	排序码	第一步	第二步	第三步	第四步	第五步	第六步	第七步	第八步	第九步
1	56	56	47	25	25	13	13	13	13	5
2	82	47	25	47	13	18	18	18	5	13
3	47	25	56	13	18	25	25	5	18	18
4	25	82	13	18	47	47	5	25	25	25
5	93	13	18	56	56	5	47	47	47	47
6	13	18	62	62	5	56	56	56	56	56
7	18	62	75	5	62	62	62	62	62	62
8	62	75	5	75	75	75	75	75	75	75
9	75	5	82	82	82	82	82	82	82	82
10	5	93	93	93	93	93	93	93	93	93

图1-1 起泡排序

[例1-1] 使用起泡排序法编程，对已知数据序列进行排序。

该程序由主函数main( )和一被调用函数bubble-sort( )组成，函数bubble-sort( )是用起泡排序法编写的排序函数。该函数有2个参数：一个是指针，它指向被排序的数据序列，即一个int型数组；另一个参数是待排序数据项的个数。

具体程序如下：

```
int a[]={56,82,47,25,93,13,18,62,75,8};
void bubble_sort();
main()
{
  register int i;
  printf("before sort : ");
  for(i=0;i<10;i++)
    printf("%d,",a[i]);
  printf("\n");
  bubble_sort(a,10);
  printf("after sort : ");
```

```
    for(i=0;i<10;i++)
      printf("%d,",a[i]);
    printf("\n");
}
void bubble_sort(s,n)
int *s,n;
{
  register int i,j;
  int temp;
  for(i=0;i<n;i++)
    for(j=n-1;j>=i;j--)
    {
      if(s[j-1]>s[j])
      {
        temp=s[j-1];
        s[j-1]=s[j];
        s[j]=temp;
      }
    }
}
```

执行该程序结果如下：

```
before sort: 56,82,47,25,93,13,18,62,75,5,
after sort: 5,13,18,25,47,56,62,75,82,93,
```

(2) 快速排序法

按照前面讲过的快速排序法的操作方法，对下列数据序列进行排序，具体执行过程如图1-2(a)—图1-2(d)所示。

待排序的数据序列如下：

56　82　47　25　93　13　18　62

该序列共有8个数据。假定左边数据序号为l，右边序号为r，选定中间一个数据项为标准数据，其序号为$\frac{1}{2}(l+r)$，在上述数据序列中标准数据为25。再从两头向内与标准项进行比较，将大于25的移到25的右边，将小于25的移到25的左边，于是形成以25为界限，左和右各为一组新的序列。对这两个新序列按此办法处理，直到排好序为止。

对上述数据序列具体操作如下：

先选25为标准项将原序列分为两个新序列，则作法如图1-2(a)所示。

56	82	47	[25]	93	13	18	62
18	82	47	[25]	93	13	56	62
18	13	47	[25]	93	82	56	62
18	13	[25]	47	93	82	56	62

图 1-2(a)

经过第一步，做了3次交换，7次比较，形成了一个由25为界限的两个数据序列：

[18 13] 25 [47 93 82 56 62]

按照上述办法处理前序列，其结果如图 1-2(b)所示。

18 13 25 [47 93 82 56 62]

13 18 25 [47 93 82 56 62]

图 1-2(b)

按照上述办法再处理新的序列，其结果如图 1-2(c)所示。

13 18 25 47 93 82 56 62

13 18 25 47 62 82 56 93

13 18 25 [47 62 56] 82 93

图 1-2(c)

再按照上述办法处理又被分出来的新序列，其结果如图 1-2(d)所示。

13 18 25 47 62 56 82 93

13 18 25 47 56 62 82 93

图 1-2(d)

该序列到此已排好。

[例 1-2] 使用快速排序法编程，对已知数据序列进行排序。

该程序由 main( )和一被调用函数 quick-sort( )组成。函数 quick-sort( )是用快速排序法编写的排序函数，该函数有 3 个参数：一个是指针，指向待排序序列；另外 2 个参数分别是待排序的序列开始和终止排序时数据项的序号。

具体程序如下：

```
int a[]={56,82,47,25,93,13,18,62};
void quick_sort();

main()
{
  register int i;
  printf("before sort : ");
  for(i=0;i<8;i++)
    printf("%d,",a[i]);
  printf("\n");
  quick_sort(a,0,7);
  printf("after sort : ");
  for(i=0;i<8;i++)
    printf("%d,",a[i]);
  printf("\n");
}

void quick_sort(s,l,r)
int *s,l,r;
{
```

```
    register int i,j;
    int x,y;
    i=l;
    j=r;
    x=s[(l+r)/2];
    do {
        while(s[i]<x&&i<r) i++;
        while(x<s[j]&&j>l) j--;
        if(i<=j)
        {
          y=s[i];
          s[i]=s[j];
          s[j]=y;
          i++;
          j--;
        }
    }while(i<=j);
    if(l<j) quick_sort(s,l,j);
    if(i<r) quick_sort(s,i,r);
}
```

执行该程序的结果如下：

```
before sort: 56, 82, 47, 25, 93, 13, 18, 62,
after sort:  13, 18, 25, 47, 56, 62, 82, 93,
```

2. 插入排序法

(1) 直接插入排序法

按照前面讲过的直接插入排序法，假定待排序数据序列共有 10 个元素，如下所示：

56 82 47 25 93 13 18 62 75 8

从 i=2 开始经过 9 步插入完成全部排序工作，具体操作过程如图 1-3 所示：

i=1	[56]	82	47	25	93	13	18	62	75	8
i=2	[56	82]	47	25	93	13	18	62	75	8
i=3	[47	56	82]	25	93	13	18	62	75	8
i=4	[25	47	56	82]	93	13	18	62	75	8
i=5	[25	47	56	82	93]	13	18	62	75	8
i=6	[13	25	47	56	82	93]	18	62	75	8
i=7	[13	18	25	47	56	82	93]	62	75	8
i=8	[13	18	25	47	56	62	82	93]	75	8
i=9	[13	18	25	47	56	62	75	82	93]	8
i=10	[8	13	18	25	47	56	62	75	82	93

图 1-3

[**例 1-3**] 使用直接插入法编程，对已知数据序列进行排序。

该程序由 main()和 insert-sort()函数组成。而 insert-sort()函数有 2 个参数：一个是指向

int 型数据序列的指针；另一个参数是该数据序列中数据项的个数。

该程序内容如下：

```
int a[]={56,82,47,25,93,13,18,62,75,8};
void insert_sort();
main()
{
  register int i;
  printf("before sort : ");
  for(i=0;i<10;i++)
    printf("%d,",a[i]);
  printf("\n");
  insert_sort(a,10);
  printf("after sort: ");
  for(i=0;i<10;i++)
    printf("%d,",a[i]);
  printf("\n");
}
void insert_sort(s,n)
int *s,n;
{
  register int i,j;
  int temp;
  for(i=1;i<n;i++)
  {
    temp=s[i];
    j=i-1;
    while(j>=0&&temp<s[j])
    {
      s[j+1]=s[j];
      j--;
    }
    s[j+1]=temp;
  }
}
```

执行该程序后结果如下：

```
before sort: 56, 82, 47, 25, 93, 13, 18, 62, 75, 8,
after sort:  8, 13, 18, 25, 47, 56, 62, 75, 82, 93,
```

(2) Shell 排序法

假设待排序数据序列有10个元素，如下所示：

56 82 47 25 93 13 18 62 75 8

按照前面讲过的 Shell 排序方法，增量取 int(n/2)，即每次增量序列长度的一半，若 n=

10，则增量分别为5，2，1。若 n＝100，则增量分别为50，25，12，6，3，1，等等。上例的具体执行过程如图1-4所示。

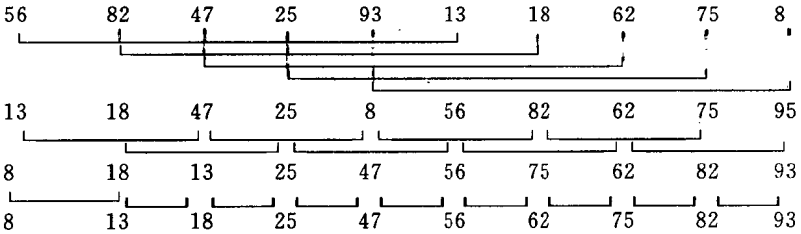

图 1-4

［例1-4］ 使用 Shell 排序法编程，对已知数据序列进行排序。

该程序由 main( )和 shell-sort( )函数组成，函数 shell_sort( )有两个参数：其中，一个是指向待排序数据的 int 型指针；另一个参数是数据序列中数据项的个数。

该程序具体内容如下：

```
int a[]={56,82,47,25,93,13,18,62,75,8};
void shell_sort();
main()
{
  int i;
  printf("before : ");
  for(i=0;i<10;i++)
    printf("%d,",a[i]);
  printf("\n");
  shell_sort(a,10);
  printf("after : ");
  for(i=0;i<10;i++)
    printf("%d,",a[i]);
  printf("\n");
}
void shell_sort(s,n)
int *s,n;
{
  register int i,j;
  int k,x;
  k=n/2;
  while(k>=1)
  {
    for(i=k;i<n;i++)
    {
      x=s[i];
      j=i-k;
```

```
        while(j>=0&&x<s[j])
        {
          s[j+k]=s[j];
          j-=k;
        }
        s[j+k]=x;
      }
      k/=2;
    }
}
```

执行该程序的结果如下：

```
before: 56, 82, 47, 25, 93, 13, 18, 62, 75, 8,
after: 8, 13, 18, 25, 47, 56, 62, 75, 82, 93,
```

3. 选择排序法

按照前面讲过的直接选择排序法的操作方法，对下列数据序列进行排序。

56 82 47 25 93 13 18 62 75 8

具体执行过程如图1-5所示。

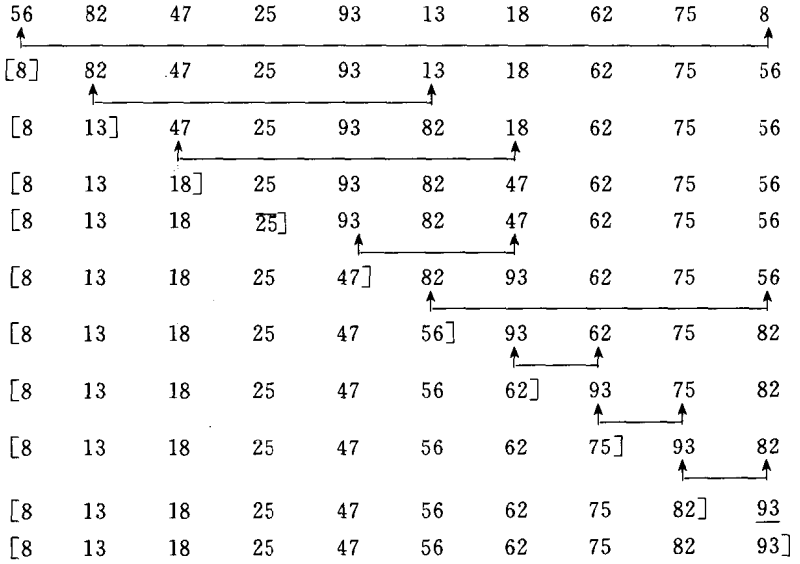

图 1-5

[例1-5] 使用直接选择排序法编程，对已知数据序列进行排序。

该程序由 main( )和 select_sort( )函数组成。select_sort( )函数是一个按直接选择排序法编写的排序函数，它有两个参数：一个是 int 型指针，指向待排序的数组首地址；另一个是待排序数据序列中数据项的个数。

该程序具体内容如下：

```
int a[]={56,82,47,25,93,13,18,62,75,8};
void select_sort();
main()
```