



西北工业大学  
NORTHWESTERN POLYTECHNICAL UNIVERSITY

---

# C程序设计 Programming in C



**1011014**

---

主讲：姜学锋，计算机学院

## 实现排序算法

- ◆ 1、排序概念
- ◆ 2、冒泡排序
- ◆ 3、选择排序
- ◆ 4、插入排序

## 6.5 数组应用程序举例

---

### ▶ 1. 排序

- ▶ 排序问题是程序设计中的典型问题，它有很广泛的应用，其功能是将一个数据元素序列的无序序列调整为有序序列，下面给出相关定义。

## 6.5 数组应用程序举例

---

- ①排序：给定一组记录的序列  $\{r_1, r_2, \dots, r_n\}$ ，其相应的关键码分别为  $\{k_1, k_2, \dots, k_n\}$ ，排序是将这些记录排列成顺序为  $\{r_{s1}, r_{s2}, \dots, r_{sn}\}$  的一个序列，使得相应的关键码满足  $k_{s1} \leq k_{s2} \leq \dots \leq k_{sn}$ （称为升序）或  $k_{s1} \geq k_{s2} \geq \dots \geq k_{sn}$ （称为降序）。

## 6.5 数组应用程序举例

---

- ▶ ②正序：待排序序列中的记录已按关键码排好序。
- ▶ ③逆序（反序）：待排序序列中记录的排列顺序与排好序的顺序正好相反。

## 6.5 数组应用程序举例

---

- ▶ 根据待排序序列的规模以及对数据处理的要求，可以采用不同的排序方法，主要有以下三类：
- ▶ ①交换类排序法：是指借助数据元素之间的互相交换实现排序的方法，例如冒泡排序法、快速排序法；
- ▶ ②选择类排序法：是指从无序序列元素中依次选择最小或最大的元素组成有序序列实现排序的方法，例如选择排序法、堆排序法；
- ▶ ③插入类排序法：是指将无序序列的元素依次插入到有序序列中实现排序的方法，例如直接插入排序法、希尔排序法。

## 6.5 数组应用程序举例

---

- ▶ (1) 冒泡排序法
- ▶ 冒泡排序法 (bubble sort) 的基本思想是通过相邻两个记录之间的比较和交换, 使关键码较小的记录逐渐从底部移向顶部 (上升), 关键码较大的记录逐渐从顶部移向底部 (沉底), 冒泡由此得名。设由  $A[1] \sim A[n]$  组成的  $n$  个数据, 冒泡排序的过程可以描述为:

6 5 3 1 8 7 2 4

## 6.5 数组应用程序举例

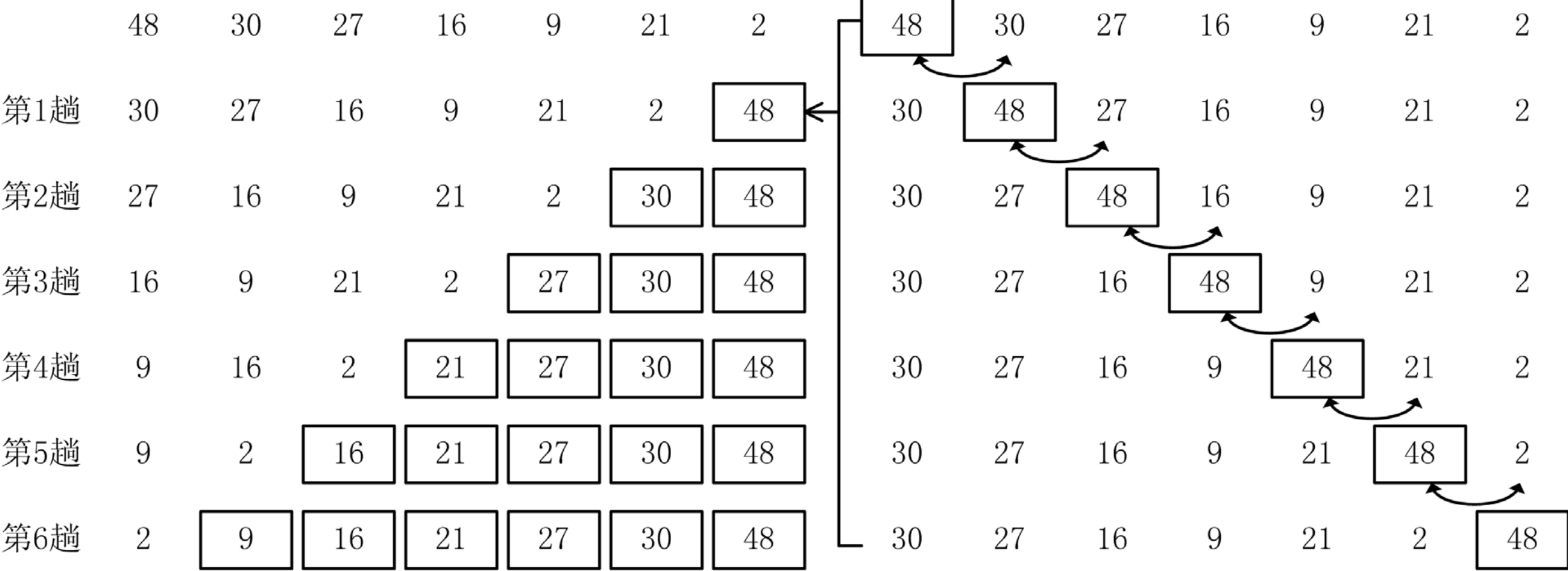
---

- ▶ ①首先将相邻的 $A[1]$ 与 $A[2]$ 进行比较，如果 $A[1]$ 的值大于 $A[2]$ 的值，则交换两者的位置，使较小的上浮，较大的下沉；接着比较 $A[2]$ 与 $A[3]$ ，同样使小的上浮，大的下沉。依此类推，直到比较完 $A[n-1]$ 和 $A[n]$ 后， $A[n]$ 为具有最大关键码的元素，称第1趟排序结束。



# 6.5 数组应用程序举例

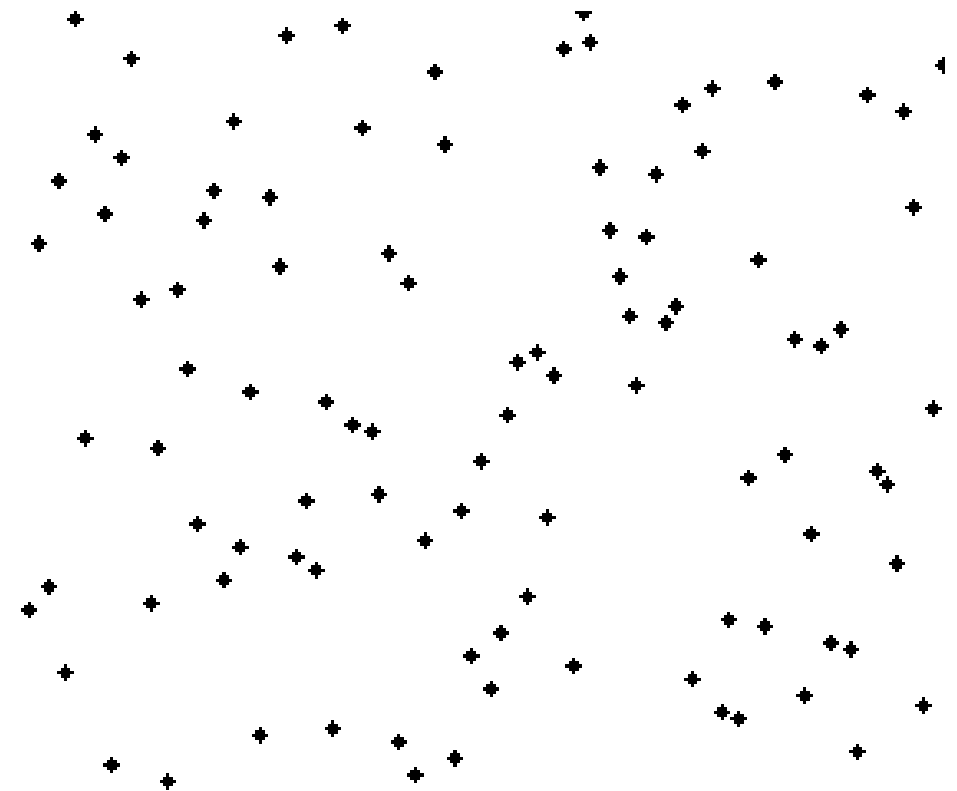
图6.9 冒泡排序



## 6.5 数组应用程序举例

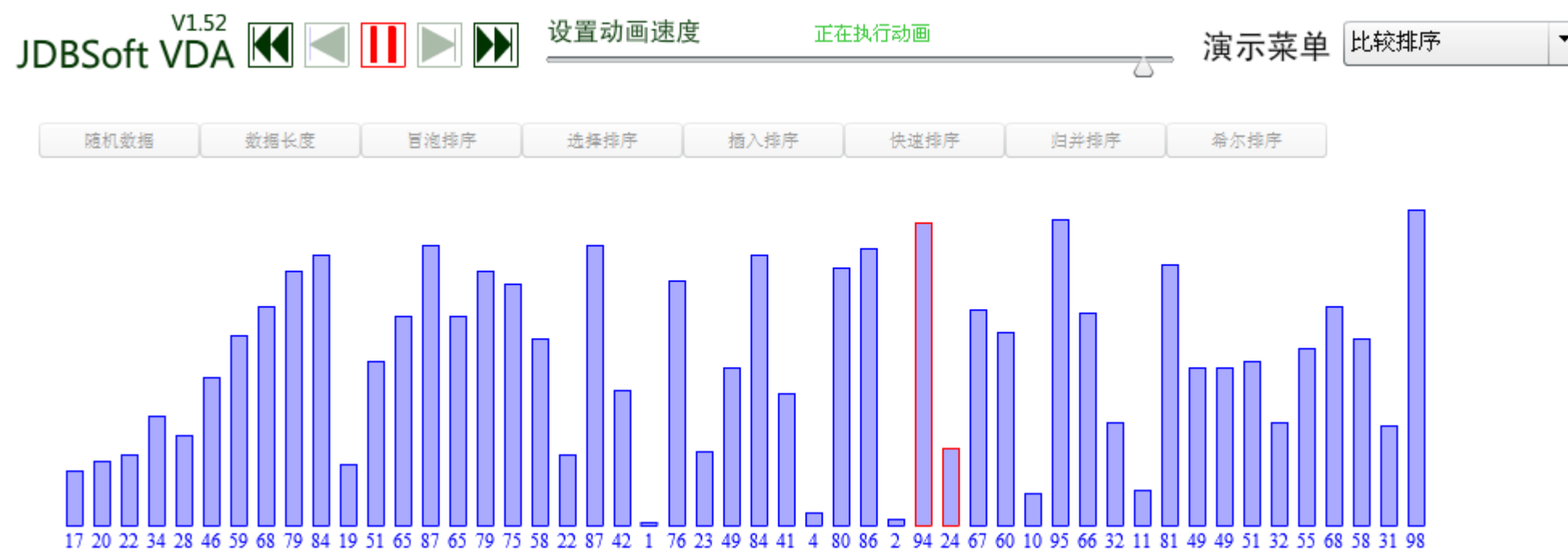
---

- ▶ ②然后在 $A[1] \sim A[n-1]$ 区间内，进行第2趟排序，使剩余元素中关键码最大的元素下沉到 $A[n-1]$ 。重复进行 $n-1$ 趟后，整个排序过程结束。



## 6.5 数组应用程序举例

### ► 冒泡排序过程演示



## 6.5 数组应用程序举例

---



### 【例6.10】

---

使用冒泡排序法将一个数组由小到大排序。

## 6.5 数组应用程序举例

### 例6.10

```
1  #include <stdio.h>
2  #define N 10 //数组元素个数
3  int main()
4  {
5      int A[N], i, j, t; //注意数组下标从0开始
6      for (i=0; i<N; i++) scanf("%d",&A[i]); //输入N个数
7      for(j=0 ; j<N-1 ; j++) //冒泡排序法
8          for(i=0 ; i<N-1-j; i++) //一趟冒泡排序
9              if(A[i] > A[i+1]) //A[i]与A[i+1]比较 <升序 >降序
10                 t=A[i], A[i]=A[i+1], A[i+1]=t; //交换
11     for (i=0; i<N; i++) //输出排序结果
12         printf("%d ",A[i]);
13     return 0;
14 }
```

## 6.5 数组应用程序举例

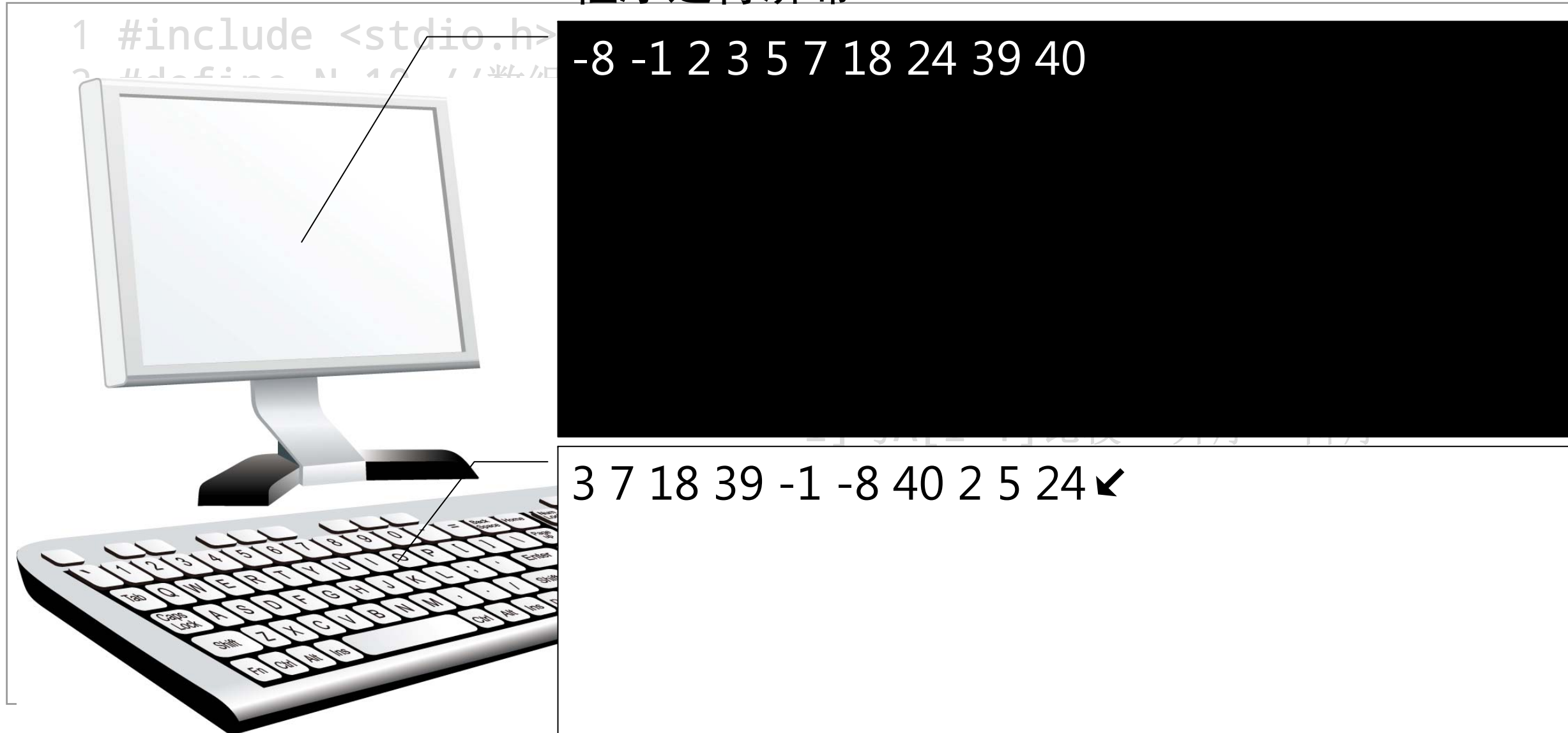
例6.10

```
1  #include <stdio.h>
2  #define N 10 //数组元素个数
3  int main()
4  {
5      int A[N], i, j, t; //注意数组下标从0开始
6      for (i=0; i<N; i++) scanf("%d",&A[i]); //输入N个数
7      for(j=0 ; j<N-1 ; j++) //冒泡排序法
8          for(i=0 ; i<N-1-j; i++) //一趟冒泡排序
9              if(A[i] > A[i+1]) //A[i]与A[i+1]比较 <升序 >降序
10                 t=A[i], A[i]=A[i+1], A[i+1]=t; //交换
11      for (i=0; i<N; i++) //输出排序结果
12          printf("%d ",A[i]);
13      return 0;
14 }
```

## 6.5 数组应用程序举例

例6.10

程序运行屏幕



## 6.5 数组应用程序举例

---

- ▶ (2) 选择排序法
- ▶ 选择排序法 (selection sort) 的基本思想是第*i*趟选择排序通过*n-i*次关键码的比较, 从*n-i+1*个记录中选出关键码最小的记录, 并和第*i*个记录进行交换。设由  $A[1] \sim A[n]$  组成的*n*个数据, 选择排序的过程可以描述为:

	8
	5
	2
	6
	9
	3
	1
	4
	0
	7



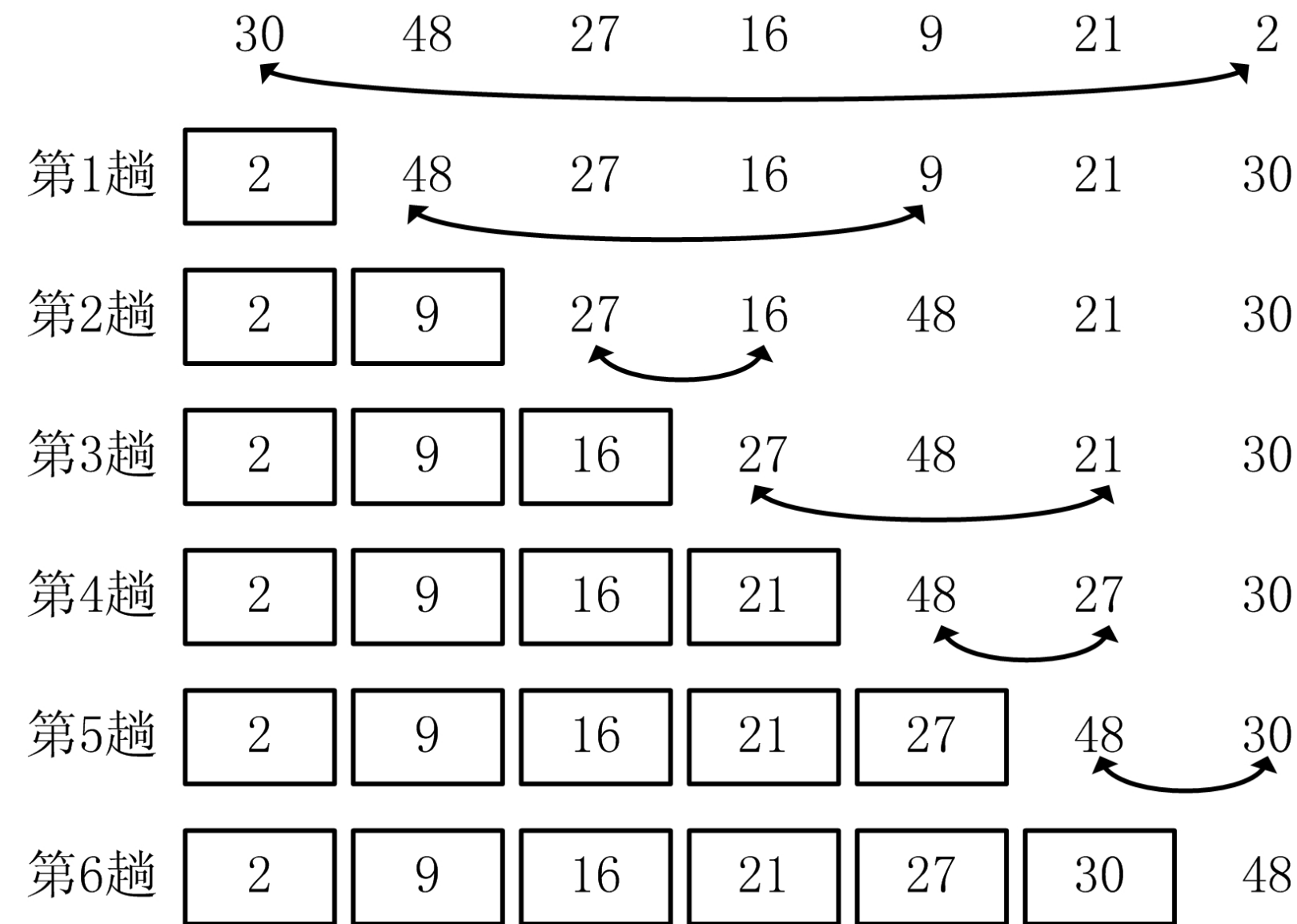
## 6.5 数组应用程序举例

---

- ▶ ①首先在 $A[1] \sim A[n]$ 区间内进行比较，从 $n$ 个记录中选出最小的记录 $A[k]$ ，若 $k$ 不为1则将 $A[1]$ 和 $A[k]$ 交换， $A[1]$ 为具有最小关键码的元素，称第1趟排序结束。

## 6.5 数组应用程序举例

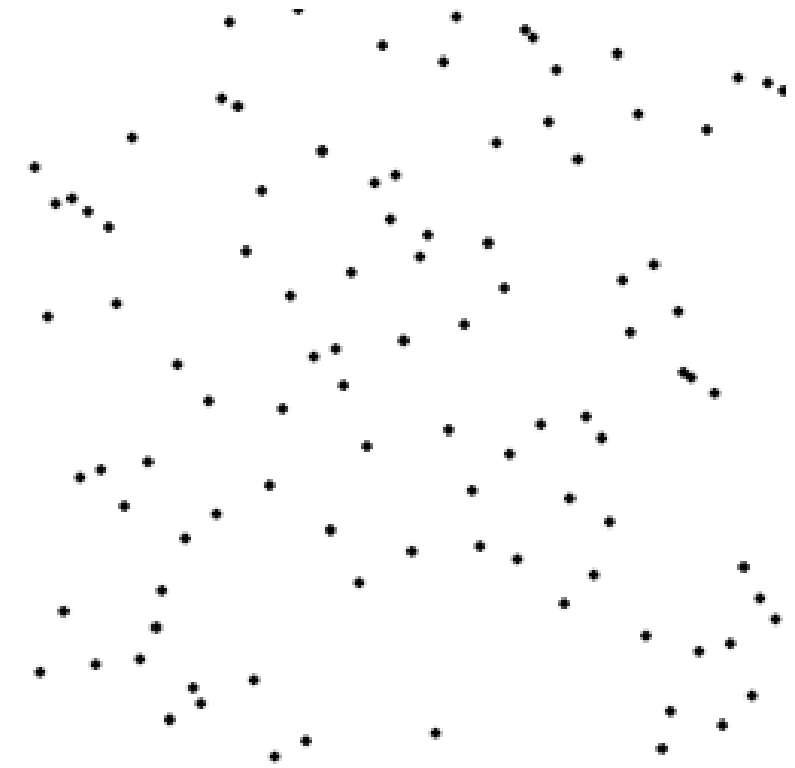
图6.10 选择排序



## 6.5 数组应用程序举例

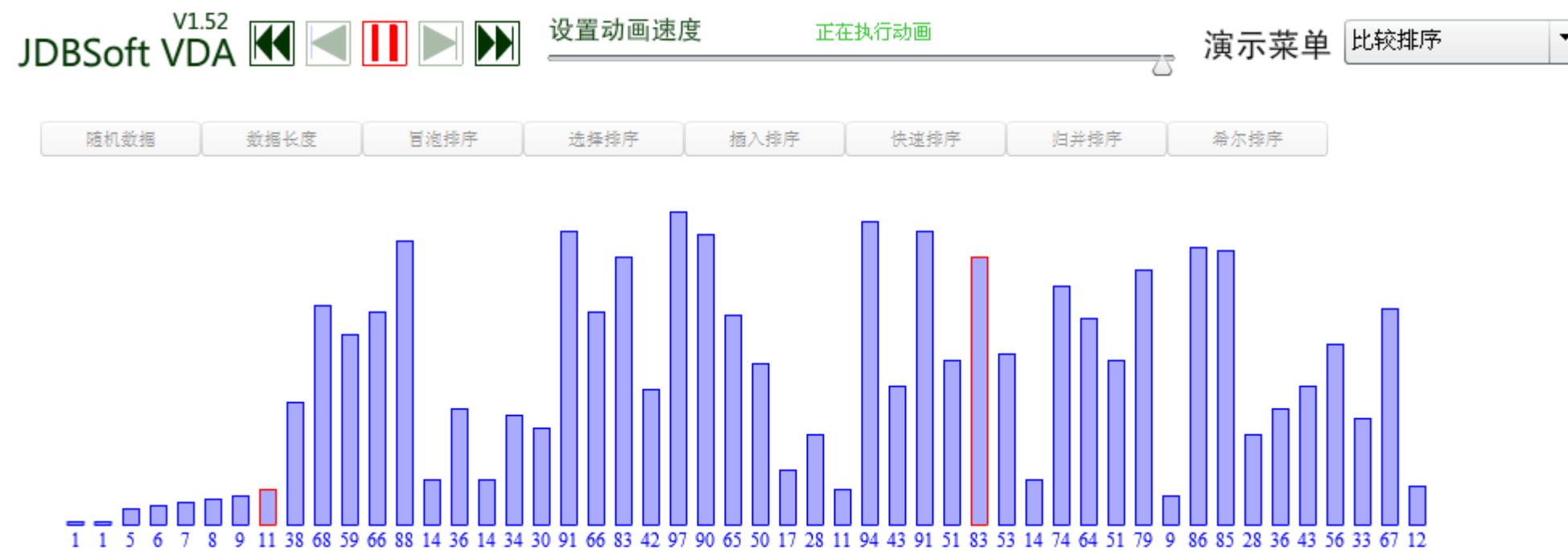
---

- ▶ ②然后在 $A[i] \sim A[n]$ 区间内，进行第 $i$ 趟排序，从 $n-i+1$ 个记录中选出最小的记录 $A[k]$ ，若 $k$ 不为 $i$ 则将 $A[i]$ 和 $A[k]$ 交换。重复进行 $n-1$ 趟后，整个排序过程结束。



## 6.5 数组应用程序举例

### ▶ 选择排序过程演示



## 6.5 数组应用程序举例

---



### 【例6.11】

---

编写选择排序函数SelectionSort，将一个数组由小到大排序。

## 6.5 数组应用程序举例

### 例6.11

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4 void SelectionSort(int A[],int n) //选择排序 n为数组元素个数
5 {
6     int i,j,k,t;
7     for(i=0; i<n-1; i++) { //选择排序法
8         k=i;
9         for(j=i+1; j<n; j++) //一趟选择排序
10             if (A[j] < A[k]) k=j; //<升序 >降序
11         if(i!=k)
12             t=A[i], A[i]=A[k], A[k]=t;
13     }
14 }
15 #define N 10
```

## 6.5 数组应用程序举例

### 例6.11

```
16 int main()
17 {
18     int A[N],i;
19     srand((unsigned int)time(0)); //设置随机数种子
20     for(i=0; i<N; i++) { //随机产生N个数
21         A[i] = rand()%100;
22         printf("%d ",A[i]);
23     }
24     SelectionSort(A,N);
25     printf("\n");
26     for(i=0; i<N; i++) printf("%d ", A[i]); //输出排序结果
27     return 0;
28 }
```

## 6.5 数组应用程序举例

例6.11

程序运行屏幕





## 6.5 数组应用程序举例

---

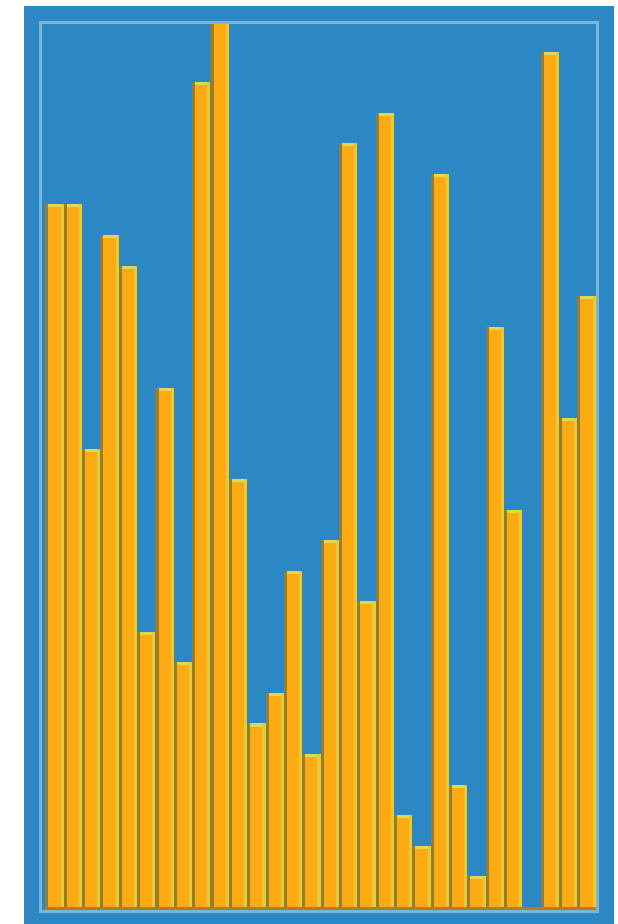
- ▶ (3) 插入排序法
- ▶ 插入排序法 (insertion sort) 的基本思想是把新插入记录的关键码与已排好序的各记录关键码逐个比较，当找到第一个比新记录关键码大的记录时，该记录之前即为插入位置k。然后从序列最后一个记录开始到该记录，逐个后移一个单元，将新记录插入k位置。如果新记录关键码比序列中所有的记录都大则插入到最后位置。

6 5 3 1 8 7 2 4

## 6.5 数组应用程序举例

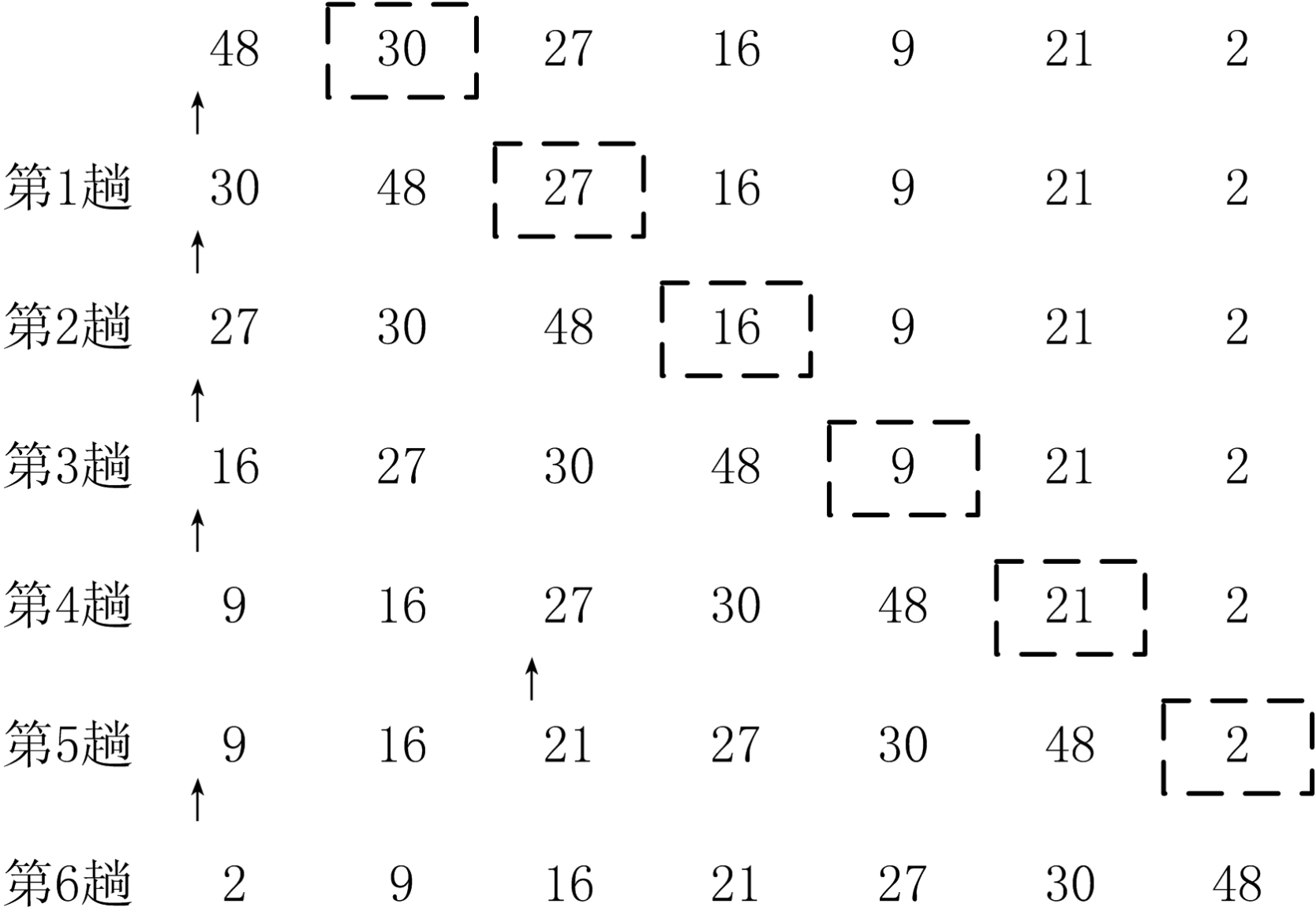
---

- ▶ 设由 $A[1] \sim A[n]$ 组成的 $n$ 个数据，插入排序的过程可以描述为：
- ▶ ①已排序列首先为 $A[1]$ 。



# 6.5 数组应用程序举例

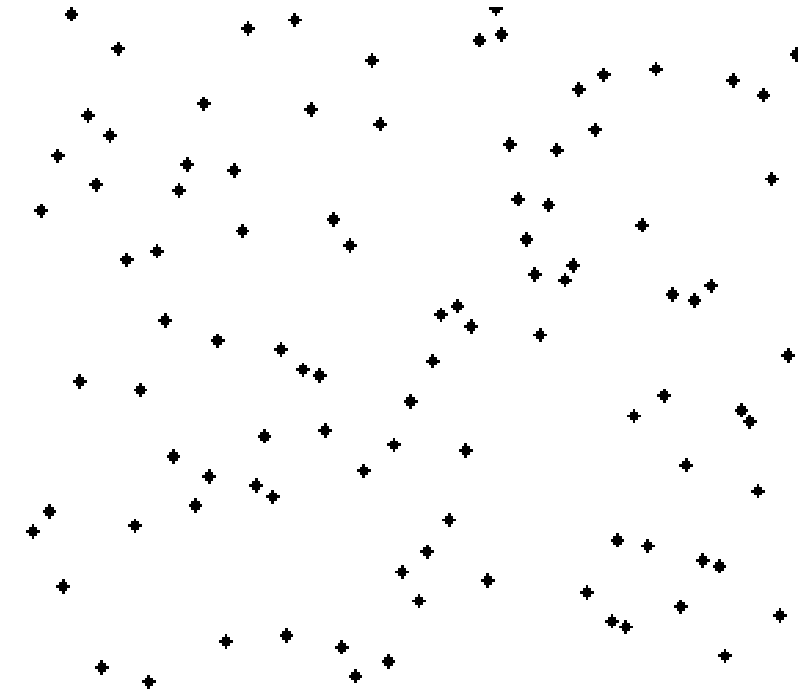
图6.11 插入排序



## 6.5 数组应用程序举例

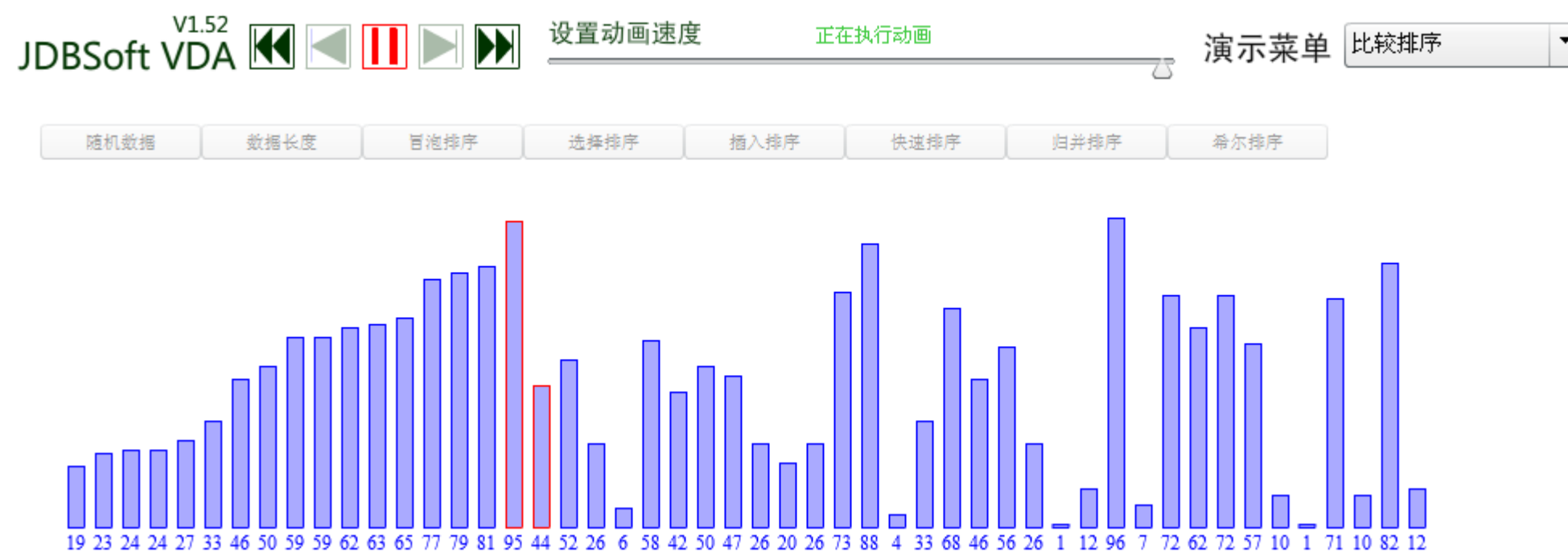
---

- ▶ ②然后将 $A[2] \sim A[n]$ 逐个插入序列中，进行第 $i$ 趟排序。将 $A[i]$ 与 $A[1] \sim A[i-1]$ 关键码进行比较，若找到 $A[k]$ 比 $A[i]$ 大，则 $A[i-1] \sim A[k]$ 逐个后移一个单元，将 $A[i]$ 插入到 $k$ 位置；若 $A[i]$ 比所有元素都大则什么也不做。重复进行 $n-1$ 趟后，整个排序过程结束。



## 6.5 数组应用程序举例

### ▶ 直接插入排序过程演示



## 6.5 数组应用程序举例

---



### 【例6.12】

---

编写插入排序函数InsertionSort，将一个数组由小到大排序。

## 6.5 数组应用程序举例

### 例6.12

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4 void InsertionSort(int A[],int n) //插入排序 n为数组元素个数
5 {
6     int i,k,t;
7     for(i=1; i<n; i++) { //插入排序法
8         t=A[i]; k=i-1;
9         while(t < A[k]) { //一趟插入排序 <升序 >降序
10             A[k+1] = A[k]; k--;
11             if(k==-1) break;
12         }
13         A[k+1]=t;
14     }
15 }
```

## 6.5 数组应用程序举例

### 例6.12

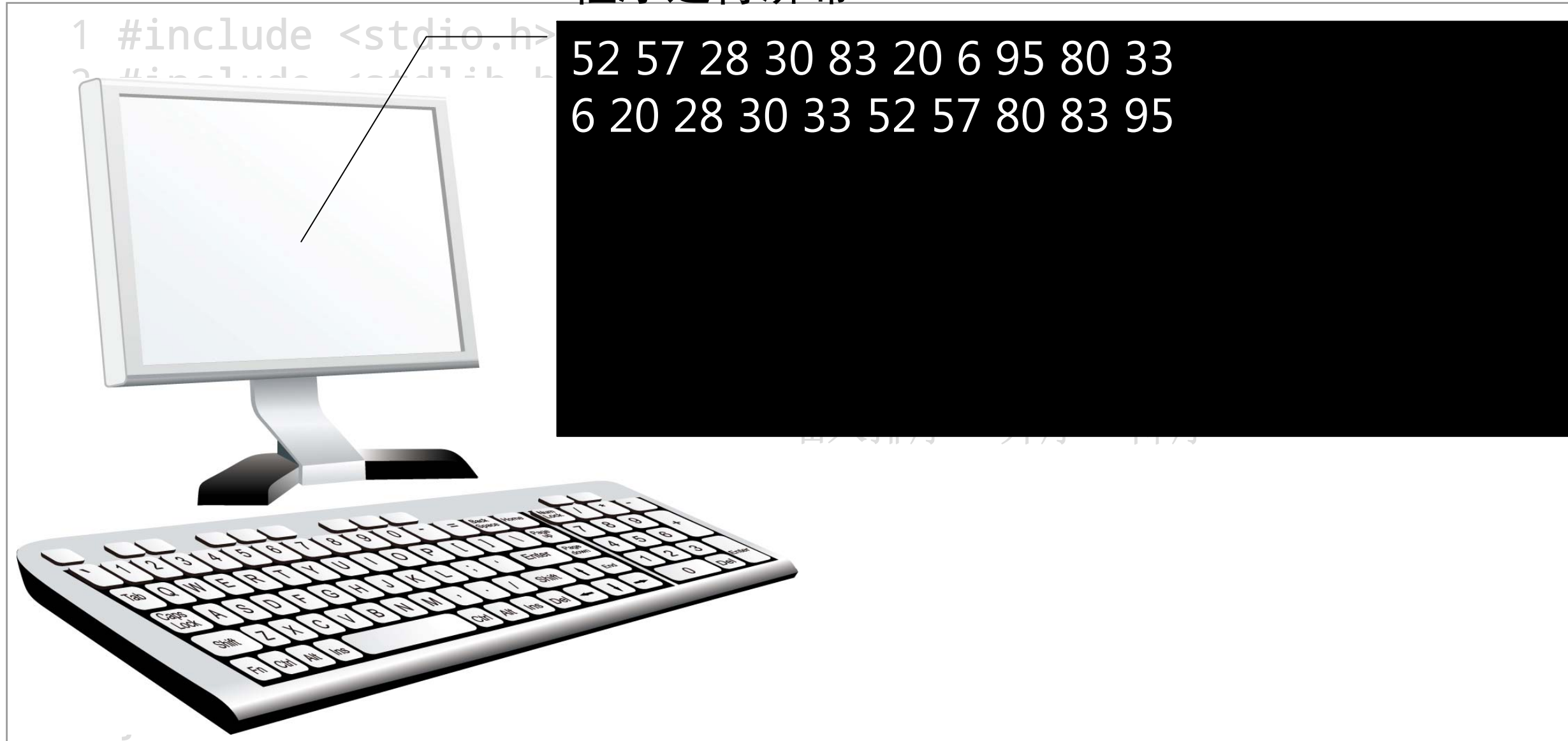
```
16 #define N 10
17 int main()
18 {
19     int A[N],i;
20     srand((unsigned int)time(0)); //设置随机数种子
21     for(i=0; i<N; i++) { //随机产生N个数
22         A[i] = rand()%100;
23         printf("%d ",A[i]);
24     }
25     InsertionSort(A,N);
26     printf("\n");
27     for(i=0; i<N; i++) printf("%d ", A[i]); //输出排序结果
28     return 0;
29 }
```



## 6.5 数组应用程序举例

例6.12

程序运行屏幕



**CP 程序设计**