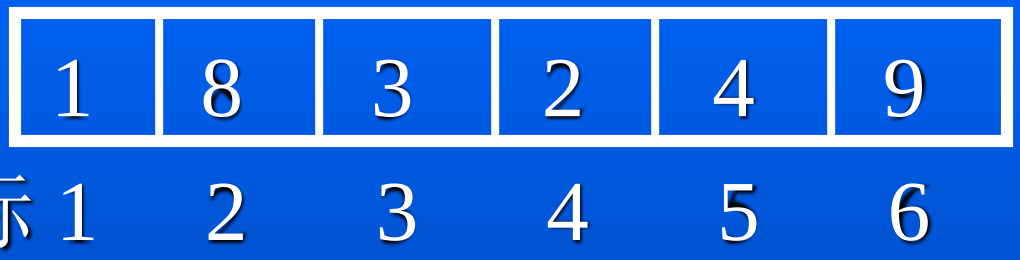
6 数据的组织与处理（一）

——数组

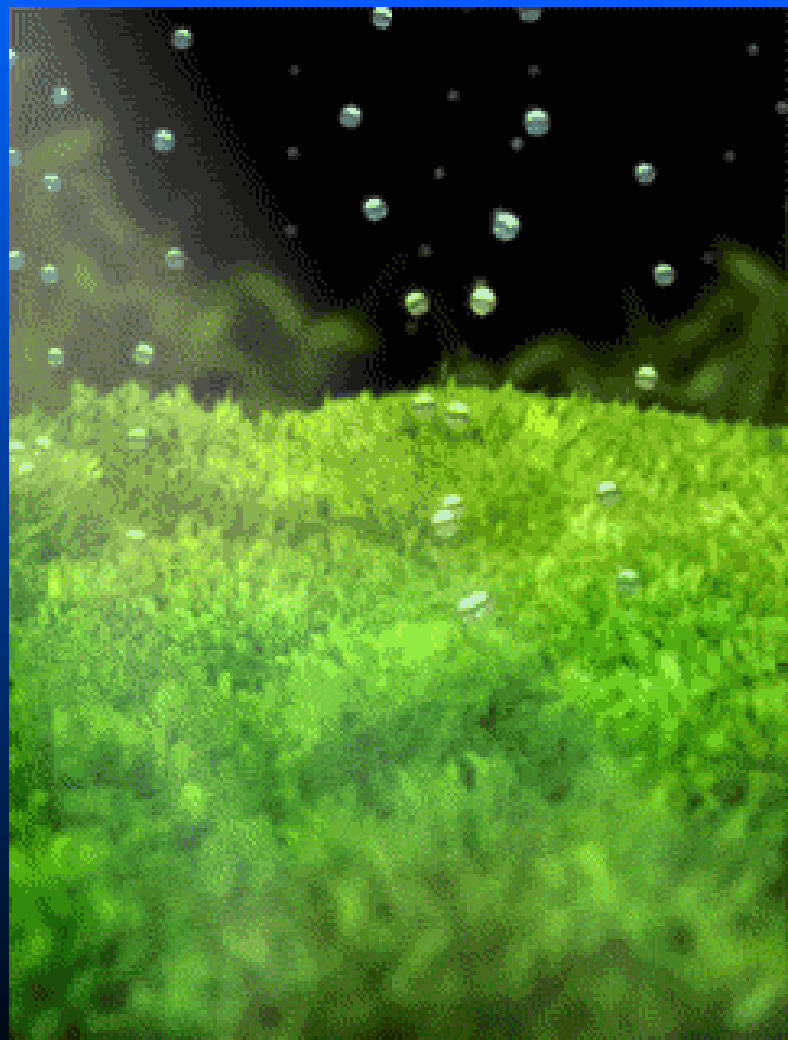
6.4 冒泡法排序

任务：　将若干个数
从大到小排序并输出

（从小到大，同理）



希望排成：



由这两幅图片，

你有何启发？

	i=1	i=2	i=3	i=4	i=5	i=6
	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
初始值	1	8	3	2	4	9
1<8；1,8互换	1 ⟷	8	3	2	4	9
1<3；1,3互换	8	1 ⟷	3	2	4	9
1<2；1,2互换	8	3	1 ⟷	2	4	9
1<4；1,4互换	8	3	2	1 ⟷	4	9
1<9；1,9互换	8	3	2	4	1 ⟷	9
1到达位置	8	3	2	4	9	1
8>3;顺序不动	8	3	2	4	9	1
3>2;顺序不动	8	3	2	4	9	1
2<4；2,4互换	8	3	2 ⟷	4	9	1
2<9；2,9互换	8	3	4	2 ⟷	9	1
2到达位置	8	3	4	9	2	1

j=1

j=2

	i=1	i=2	i=3	i=4	i=5	i=6	
	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	
中间结果	8	3	4	9	2	1	
8>3;顺序不动	8	3	4	9	2	1	
3<4； 3, 4互换	8	3 ⟷	4	9	2	1	j=3
3<9； 3, 9互换	8	4	3 ⟷	9	2	1	
3到达位置	8	4	9	(3)	2	1	
8>4;顺序不动	8	4	9	3	2	1	
4<9； 4, 9互换	8	4 ⟷	9	3	2	1	j=4
4到达位置	8	9	(4)	3	2	1	
8<9； 8, 9互换	8 ⟷	9	4	3	2	1	j=5
8到达位置	9	(8)	4	3	2	1	

冒泡排序算法分析：

从表中可以看出最小的一个数第一遍扫描就交换到a[6]中。如将a[1]视为水底，a[6]视为水面：

- 最”轻”的(最小的)一个数 1 最先浮到水面，交换到a[6];
- 次”轻”的 2 第二遍扫描交换到a[5];
- 再”轻”的 3 第三遍扫描交换到a[4];

…

依此类推，有6个数，前5个数到位需5遍扫描，此时，第6个最”重”的数自然已经落在了a[1]中。因此，6个数实际上只需要5遍扫描就可以了。

若令 扫描总遍数为 j，则显然有：j=n-1。

	扫描遍数 j									
	1		2		3		4		5	
变量i	从	到	从	到	从	到	从	到	从	到
	1	5	1	4	1	3	1	2	1	1

n=6　　n-1　　n-2　　n-3　　n-4　　n-5

j 从 1 到 5（即n-1）

→ 则i从1到 n-j

n-j

找规律

构思扫描程序

```
for ( j=1; j<=n-1; j=j+1)
    for ( i=1; i<=n-j; i=i+1)
        a[i] 与 a[i+1]
              比较，交换
```

冒泡排序算法设计：

为了表述方便，定义以下3个变量：

- n —— 待排序的数的个数，这里 n=6
- j —— 扫描遍数，j=1,2,…,n-1
- i —— 第j遍扫描待比较元素的下标，i=1,2,…,n-j

采用两重计数型循环：

- 步骤1：将待排序的数据放入数组中；
- 步骤2：置j为1；
- 步骤3：让 i 从1到n-j，比较a[i]与a[i+1]，
  如果 a[i] >= a[i+1]，位置不动；
  如果 a[i] < a[i+1] ，位置交换，
- 步骤4：让 j = j+1；只要 j != n
  返回步骤3 ，当j==n 时执行步骤5
- 步骤5：输出排序结果

```
//**************************************
//* 程 序 名：5_4.cpp                  *
//* 作      者：wuwh                   *
//* 编制时间：2002年9月22日            *
//* 主要功能：冒泡排序                 *
//**************************************
#include <iostream>// 预编译命令
#include <memory>// 预编译命令
using namespace std;
```

为什么数组变量a的大小设为 7？

```
int  main()
{
    int i = 0, j = 0, p = 0, a[ 7 ];

    memset( a, 0, sizeof( a ) );

    for ( i =1; i <=6; i++ )

    {
        cout  << "请输入待排序的数a[“
              << i << “ ]= “;
        cin >> a [ i ];
     }
```

```
for ( j=1; j<=5; j++)        // 外层循环
    for ( i=1; i<=6-j; i++ )  // 内层循环
        if ( a[i] < a[i+1] )
        {
            p = a[i];
            a[i] = a[i+1];
            a[i+1] = p;
        }
```

T

F

```
for ( j = 1;  j <= 5;  j++ )          // 外层循环
   for ( i = 1; i <= 6-j; i++ )       // 内层循环
      if ( a[ i ] < a[ i+1 ] )
       // 让 a[i] 与 a[i+1] 交换
       {
              p = a[ i ];
              a[ i ] = a[ i+1 ];
              a[ i+1] = p;
        }
```

```
for ( i = 1; i <=6 ; i++)
    // 输出排序结果
    cout << a[ i ] << endl;
  return 0;
}
```

从小到大的冒泡排序示意

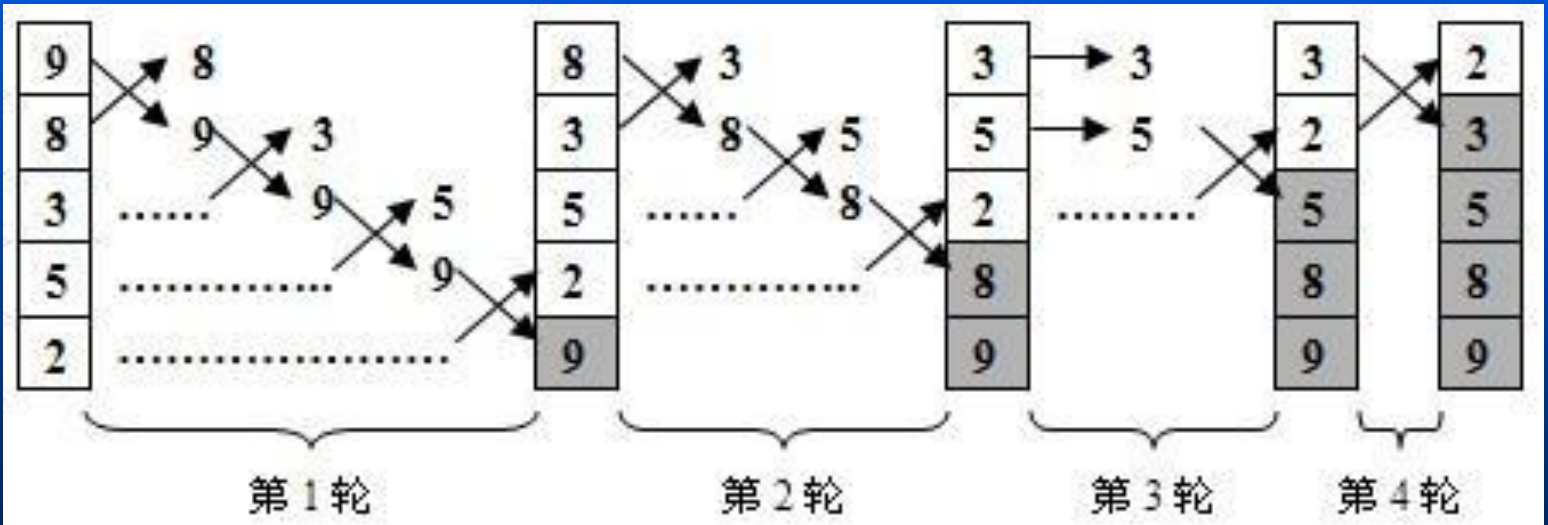


图 冒泡排序过程

关于冒泡法的几个思考题

- 如何将程序改为将 0 作为数组的第一个元素?
- 如果数组规模扩大了，有100个元素、1000个元素、等等，程序又应该改动哪些地方?
- 如果数组的元素数目在程序运行后才能确定（由用户从键盘输入），程序又应该如何修改?
- 如果不是从大到小排列，而是希望从小到大排列，程序应该如何修改?

关于冒泡法的几个思考题

- 如果排序规则在程序运行时确定（请考虑如何输入排序的规则），程序应该如何修改?
- 如果数组中不是整数类型的元素，而是其他类型，程序又应该如何修改？或者，你希望能怎样修改更方便?
- 如果数组元素很多，程序运行的速度如何?能更快一些吗?
- 为什么要排序呢?

结　束