冒泡排序

【问题描述】

输入 n（n≤100）个正整数，然后按从小到大的顺序输出。

【输入样例】

```
10
2 5 8 6 12 34 65 22 16 55
```

【输出样例】

```
2 5 6 8 12 16 22 34 55 65
```

【分析】

下面用 n=5 个数举例来说明冒泡排序的核心思想:

第一轮：

3 2 4 5 1	3 和 2 比较，交换位置，排成下行的顺序;
2 3 4 5 1	3 和 4 比较，不交换，维持同样的顺序;
2 3 4 5 1	4 和 5 比较，不交换，顺序不变;
2 3 4 5 1	5 和 1 比较，交换位置，排成下行的顺序;
2 3 4 1 5	经 1~ (n-1)次比较后，将 5 调到了末尾。

经过第一轮的 1~(n-1)次比较，就能把 n 个数中的最大数调到最末尾位置。

第二轮：

2 3 4 1 5	2 和 3 比较，不交换，维持同样的顺序;
2 3 4 1 5	3 和 4 比较，不交换，维持同样的顺序;
2 3 4 1 5	4 和 1 比较，交换位置，排成下行的顺序;
2 3 1 4 5	经 1~ (n-2)次比较后，将 4 和 5 调到了末尾。

经过第二轮 1~(n-2)次比较，又把这一轮 n-1 个数中的“最大数”调到这 n-1 个数的“最末尾”位置。

第三轮：

2 3 1 4 5　　2 和 3 比较，不交换，维持同样的顺序；

2 3 1 4 5　　3 和 1 比较，交换位置，排成下行的顺序；

2 1 3 4 5　　经 1~ (n-3)次比较后，将 3、4 和 5 调到了末尾。

经过第三轮 1~(n-3)次比较，又把这一轮 n-2 个数中的“最大数”调到这 n-2 个数的“最末尾”位置。

第四轮：

2 1 3 4 5　　2 和 1 比较，交换位置，排成下行的顺序；

1 2 3 4 5　　经 1~ (n-4)次比较后，将 2、3、4 和 5 调到了末尾。

经过第四轮 1~(n-4)次比较，又把这一轮 n-3 个数中的“最大数”调到这 n-3 个数的“最末尾”位置。

总之：

每进行一轮两两比较后，下一轮的比较范围就减少一个。最后一轮仅有一次比较。在比较过程中，每次都有一个“最大数”往下“掉”，用这种方法排列顺序，常被称为“冒泡”排序。

【算法流程】

1) 用循环把十个数输入到 a 数组中。

2) 从 a[0]到 a[9]，相邻的两个数两两比较，即：a[0]与 a[1]比，a[1]与 a[2]比，……，a[8]与 a[9]比。只需知道两个数中的前面元素的标号，就可以与后一个相邻的元素比较。写成通用形式就是 a[i]与 a[i+1]比较。

3) 比较的次数可以用 1~(n-i)次循环进行控制。也就是说，循环次数与两两比较时的前面元素的序号有关。

4) 在每次的比较中，若较大的数在前面，就把前后两个对换，把较大的数调到后面，否则不需调换位置。

【参考程序】

```cpp
#include<iostream>
#include<algorithm>
using namespace std;
const int N = 100;
int a[N];
int main()
{
    int n;
    cin >> n;
    //输入 n 个数
    for (int i=0; i<n; i++)
    {
        cin >> a[i];
    }
    //冒泡排序
    for (int i=1; i<n; i++)
    {
        for (int j=0; j<n-i; j++)
        {
            //两两相比较
            if (a[j] > a[j+1])
            {
                //比较与交换
                swap(a[j], a[j+1]);
            }
        }
    }
    //输出排序后的数
    for (int i=0; i<n; i++)
    {
        cout << " " << a[i];
    }
    return 0;
}
```