

C 冒泡排序算法改进篇 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/470/2021_2022_C___E5_86_92_E6_B3_A1_E6_c67_470115.htm 排序是在程序设计中常碰到的问题，排序算法也有很多种。起泡法是众所周知的排序算法，其原理是每次将相邻两个数进行比较，较大的下沉。其的主程序段如下（用VC实现）：

```
Void Bubble Sort (int* pData,int Count) { Int iTemp. for(int i=1.i { For (int j=Count-1.j>=i.j--) { if(pData[j] { iTemp = pData[j-1]. pData[j-1] = pData[j]. pData[j] = iTemp. } } } }
```

我们分析上述程序段可以发现起泡法是从一端开始比较的，第一次循环就是把最小数上升到第一位置，第二次循环就是把第二最小数上升到第二位置。如此循环实现数据的排序。那么我们是否可以找到最小数的同时找到最大数呢？当然可以。方法是在一端起泡时同时在另一端也进行起泡。即反向起泡。下面的程序段实现的是双向起泡：

```
void Bubble2Sort(int* pData,int Count) { int iTemp. int left = 1. int right =Count -1. int t. do { //正向的部分 for(int i=right.i>=left.i--) { if(pData[i] { iTemp = pData[i]. pData[i] = pData[i-1]. pData[i-1] = iTemp. t = i. } } left = t 1. //反向的部分 for(i=left.i { if(pData[i] { iTemp = pData[i]. pData[i] = pData[i-1]. pData[i-1] = iTemp. t = i. } } right = t-1. }while(left}
```

分析上面的程序段我们可以发现正向起泡时第一次循环找出了最小数，反向起泡第一次循环找到最大数。很显然在一次循环中即可以找到最小的数还可以找到一个最大的数，所以用双向冒泡排序的交换的次数减少了，从而达到了优化起泡法的作用。
100Test 下载频道开通，各类考试题目直接下载。详细请访

