

昨日写完冒泡排序，和大多数人的感觉一样，太简单，丝毫没有挑战性。但楼主是一个追求踏实平稳的人，希望地基牢固，也为方便后面学习和研究更加高深的算法。但在研究效率上还有待提高，楼主一定好好努力。今天将会写完选择排序 和 插入排序，本文主在选择排序。

一. 算法描述

选择排序：比如在一个长度为N的无序数组中，在第一趟遍历N个数据，找出其中最小的数值与第一个元素交换，第二趟遍历剩下的N-1个数据，找出其中最小的数值与第二个元素交换.....第N-1趟遍历剩下的2个数据，找出其中最小的数值与第N-1个元素交换，至此选择排序完成。

以下面5个无序的数据为例：

56 12 80 91 20（文中仅细化了第一趟的选择过程）

第1趟：12 56 80 91 20

第一趟遍历选择最小数并交换

56

12 80 91 20 | 最小数值索引:1

56

12

 80 91 20 | 最小数值索引:2

56 12

80

 91 20 | 最小数值索引:2

56 12 80

91

 20 | 最小数值索引:2

56 12 80 91

20

 | 最小数值索引:2

12

56

 80 91 20 | 位置2与位置1数据互换

PS：索引即为数据元素的位置

第2趟：12 20 80 91 56

第3趟：12 20 56 91 80

第4趟：12 20 56 80 91

二. 算法分析

平均时间复杂度：O(n²)

空间复杂度：O(1) (用于交换和记录索引)

稳定性：不稳定（比如序列【5， 5， 3】第一趟就将第一个[5]与[3]交换，导致第一个5挪动到第二个5后面）

三. 算法实现

[cpp]

01. //交换data1和data2所指向的整形

02. void DataSwap(int* data1, int* data2)

03. {

04. int temp = *data1;

```
05.     *data1 = *data2;
06.     *data2 = temp;
07. }
08.
09. /*****
10. *函数名称: SelectionSort
11. *参数说明: pDataArray 无序数组;
12. *          iDataNum为无序数据个数
13. *说明:     选择排序
14. *****/
15. void SelectionSort(int* pDataArray, int iDataNum)
16. {
17.     for (int i = 0; i < iDataNum - 1; i++)    //从第一个位置开始
18.     {
19.         int index = i;
20.         for (int j = i + 1; j < iDataNum; j++)    //寻找最小的数据索引
21.             if (pDataArray[j] < pDataArray[index])
22.                 index = j;
23.
24.         if (index != i)    //如果最小数位置变化则交换
25.             DataSwap(&pDataArray[index], &pDataArray[i]);
26.     }
```

loa