打擂台算法与打擂台排序

# 打擂台算法简单介绍

打擂台算法应用于求数组中的最大值或最小值比较常见。当然也可以用于数组排序，这种方法性能较差，一般排序不采用该方法。

**打擂台排序是比较简单的排序方法**。

问题：设有一数组，其大小为10个元素**（int a[10]）**数组内的数据是无序。现在要求我们通过编程将这个无序的数组变成一个从小到大排序的数组（从下标为0开始）

思路：按照题目的要求，毫无疑问，正确的结果应该就像这样：**1 2 3 4 5 6 7 8 9 10**   要做到这样，**最简单和最直接想到的方法就是进行对比交换**。

# 算法原理：

算法描述：擂台上怎样决定出最后的优胜者？

先找一个人站在台上，第二个人上去与之比武，获胜者留在擂台上。再上去第三个人，与台上的人（即刚才的得胜者）比武，胜者留在台上，败者下台，直到所有的人都比试过后，最后留在擂台上的即是冠军！

应用：**求一组数据中的最大值或者最小值。**

**打擂台排序：可以看做N-1次打擂台算法，每一次选出一名最优的，直到结束。**

首先选择索引最小的位置作为擂台，然后与剩下的其他元素依次做比较，若其他元素比该元素值小，则交换元素值，这样一轮下来，就确保了索引为0的位置上（即擂台上）是最小的值。其次，依次选择索引为1、2、…N-2的位置作为擂台。（N-1就没有必要作为擂台了）。

# Java代码：

// 打擂台获取最大值

**public** **static** **int** daLeiTaiMax(**int** a[]){

**int max = a[0];**

**for**(**int** j = 1;j < a.length;j++){

**if**(max < a[j]){

max = a[j];

}

}

**return** max;

}

// 打擂台获取最小值

**public** **static** **int** daLeiTaiMin(**int** a[]){

**int min = a[0];**

**for**(**int** j = 1;j < a.length;j++){

**if**( a[j]<min){

min = a[j];

}

}

**return** min;

}

// 打擂台排序（相当于**打N-1次擂台**）

**public** **static** **void** daLeiTaiSort(**int** a[]){

**int** N = a.length;

**int** temp = 0;

**for**(**int** i = 0;i< N-1;i++){

**for**(**int** j = i+1;j < N;j++){

**if**(a[i] > a[j]){

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

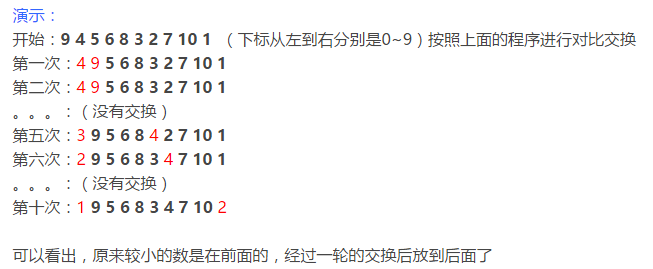
}

}

# 打擂台排序存在的不足

打擂台是比较容易想到的实现方法。但是存在不足：**就是位于前面的较小的数被交换到后面**。

**不足之处演示**：



原来在前面的较小的数，经过一轮之后交换到后面，这样在接下来的轮数中，会再将较小的数交换到前面，带来的不必要的操作。

**那么怎样解决这个不足呢？可以使用冒泡排序**

什么是冒泡排序呢？

你可以这样理解：（从小到大排序）存在10个不同大小的气泡，由底至上地把较少的气泡逐步地向上升，这样经过遍历一次后，最小的气泡就会被上升到顶（下标为0），然后再从底至上地这样升，循环直至十个气泡大小有序。

**在冒泡排序中，最重要的思想是两两比较，将两者较少的升上去。**

**冒泡排序最坏情况的时间复杂度是O(n²)**。

比较次数为：**n\*（n-1）/2**;交换次数最大为**3n\*（n-1）/2。**