冒泡排序（Bubble Sort）

一、原始算法

1. 概念：在要排序的一组数中，对当前还未排好序的范围内的全部数，自上而下对相邻的两个数依次进行比较和调整，让较大的数往下沉，较小的往上冒。即：每当两相邻的数比较后发现它们的排序与排序要求相反时，就将它们互换。
2. 方法：类似于1与2比较—>2与3比较—>3与4比较……这样一直下去，每一趟都会让一趟中的最大值沉下去，由于N个数这样比较在最坏的情况下需要N-1次比较所以最外层：



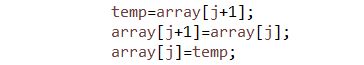
而内层的交换比较初始时需比较N-1次，而后每次只需要比较N-1-1次这里的后面的那个1，我们通过外层的循环i控制。



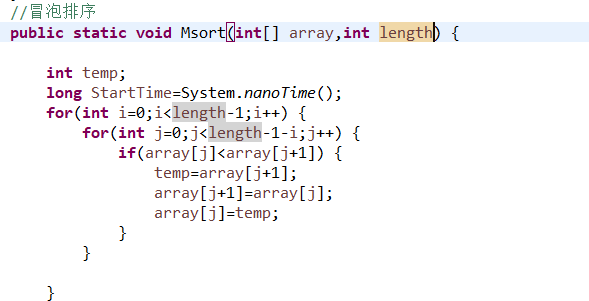
至于升序降序就根据你是让大数下沉还是让小数下沉了，这取决if判断，如果数组后面的数大于前面的数的话，这时交换就是降序，如例，否则反之。



之后的值的交换交给临时值temp,用来临时存储值，帮助交换。



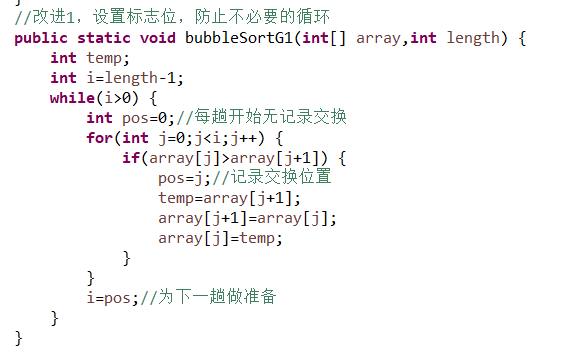
1. 例子(降序)：



二、改进算法

冒泡排序还可以改进使它的运行效率大大提高。

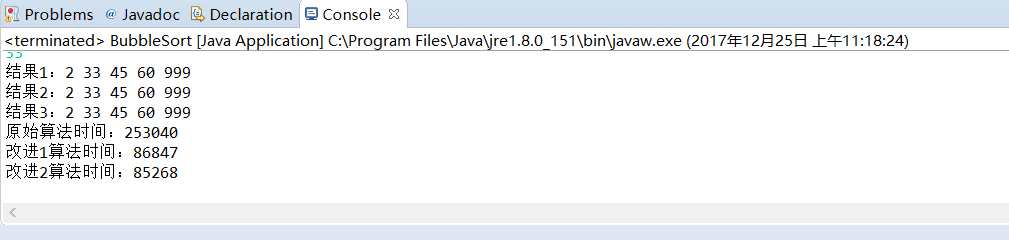
1. 对冒泡排序常见的改进方法是加入一标志性变量exchange，用于标志某一趟排序过程中是否有数据交换，如果进行某一趟排序时并没有进行数据交换，则说明数据已经按要求排列好，可立即结束排序，避免不必要的比较过程。设置一标志性变量pos,用于记录每趟排序中最后一次进行交换的位置。由于pos位置之后的记录均已交换到位,故在进行下一趟排序时只要扫描到pos位置即可。



1. 传统冒泡排序中每一趟排序操作只能找到一个最大值或最小值,我们考虑利用在每趟排序中进行正向和反向两遍冒泡的方法一次可以得到两个最终值(最大者和最小者) , 从而使排序趟数几乎减少了一半。



1. 三种方式运行结果：



从中我们可以看出运行所花费的时间在逐渐递减，最后一种方式的效率最高。