

算法设计与分析实验报告

实验名称： 算法时间复杂度的实验测试

一、问题陈述，相关背景、应用及研究现状的综述分析

实验内容： 对于一个算法（可以选择排序算法，如堆排序和插入排序），测试该算法在不同输入规模下的运行时间；

选择了俩排序，一个是冒泡排序，一个是快速排序。选择这两个算法的原因是因为他们的时间复杂度是不一样的，可以对比来看不同时间复杂度的算法在不同规模下的运行时间。

二、模型拟制、算法设计和正确性证明

用数组来储蓄要进行排序的数据。用随机数生成原始数组，对同一原始数组计算两种算法的消耗时间。用 `clock_t` 类来计算程序运行的时间。具体的时间算法就是在进行算法前用 `clock()` 函数记录一个系统时间，在算法结束后记录一个系统时间，两系统时间差就是程序运行的时间。

冒泡排序很简单，两个循环就完成了。但是由于吗，冒泡排序比快速排序的时间复杂度高，所以在一定输入规模下，冒泡排序消耗的时间过长，而快速排序消耗的时间还很短。所以限制了冒泡排序算法最多用 5 秒时间，如果超过了 5 秒时间，就主动结束冒泡排序算法。

快速排序用递归实现，用的是黄同学的思路。`Quick(left, right)` 表示在区间 `[left, right]` 中对排序数组 `tmp[]` 进行快速排序：

递归出口：`left >= right`;

取 `tmp[left]` 为基准量，计算出其应该在 `[left, right]` 的实际位置 `pos`，并且把 `[left, right]` 内的数进行整序，使得 `[left, pos-1]` 的数小于 `tmp[left]`，`[pos+1, right]` 的数大于等于 `tmp[left]`;

递归向下计算 `Quick(left, pos-1)` 和 `Quick(pos+1, right)`;

三、时间和空间复杂性分析

冒泡排序：算法的基本操作是对 $\text{tmp}[j]$ 和 $\text{tmp}[j-1]$ 的比较和可能作出的交换，一共进行了 $n*(n-1)/2$ 次，所以时间复杂度是 $O(n*n)$ ，空间复杂度是 $O(n)$ ，因为用了长度为 n 的数组储存数据。

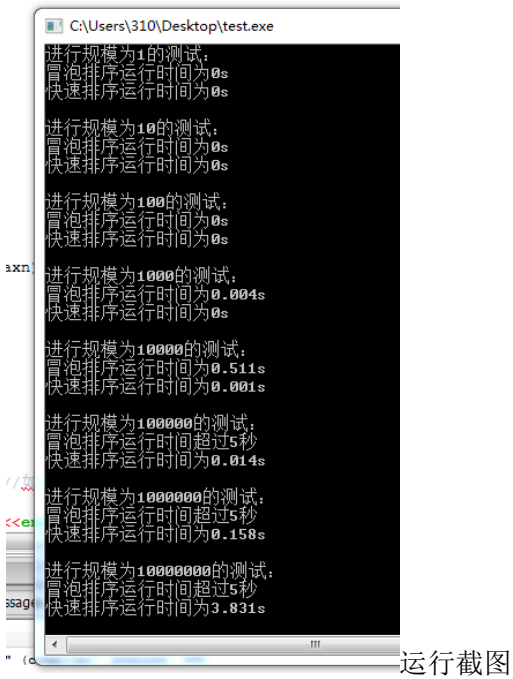
快速排序：算法的基本操作是对 $\text{tmp}[]$ 数组进行分割和在整序过程中涉及到的数组元素比较，其中每一层的整序长度是 n 的，所以会进行 n 次比较，算法会平均会分割出 $\log n$ 层出来，所以会一共比较 $n*\log n$ 次，所以时间复杂度是 $O(n*\log n)$ ，空间复杂度是 $O(n)$ ，因为用了长度为 n 的数组储存数据。

四、程序实现和实验测试过程

程序分别对输入规模为 1、10、100、1000、10000、100000、1000000、10000000 进行了测试。结果如下：

单位：s

输入规模	冒泡排序	快速排序
1. E+00	<0. 001	<0. 001
1. E+01	<0. 001	<0. 001
1. E+02	<0. 001	<0. 001
1. E+03	0. 004	<0. 001
1. E+04	0. 511	0. 001
1. E+05	>5	0. 014
1. E+06	>5	0. 158
1. E+07	>5	3. 831



运行截图

五、总结

可以从得到的运行结果看出，此电脑的每秒可以进行基本操作（比较两数组元素大小）200000000 次。