

以下对100个数据进行比较排序不同排序消耗的时间（精确到毫秒）：

插入排序0ms

希尔排序0ms

以下对1000个数据进行比较排序不同排序消耗的时间（精确到毫秒）：

插入排序0ms

希尔排序0ms

以下对10000个数据进行比较排序不同排序消耗的时间（精确到毫秒）：

插入排序266ms

希尔排序0ms

以下对100000个数据进行比较排序不同排序消耗的时间（精确到毫秒）：

插入排序17265ms

希尔排序47ms

以下对1000000个数据进行比较排序不同排序消耗的时间（精确到毫秒）：

插入排序2.09385e+006ms

希尔排序754ms

结论：

本次实验选择了两种排序方法，分别为：插入排序，希尔排序。

排序	100	1k	10k	100k	1M
插入（ms）	0	0	266	17265	2.09385e+0.06
Shell（ms）	0	0	0	47	754

插入排序的时间复杂度为 $\Theta(n^2)$, shell 排序的时间复杂度 $\Theta(n^{1.5})$. 插入排序在处理较大数据时明显用时很长，本实验运行了将近 40 分钟才运行完，所以插入排序性能相比于希尔排序，归并排序等等要差一些。希尔排序在 n 中等以及较大规模的数据进行排序时时间优势比较明显。

日志：

2019 年 6 月 13 日

本实验要求在时间复杂度 n^2 的里面选择一个，在其余的算法中选择一个，我选择的插入排序和希尔排序，实际上希尔排序不是性能最好的排序算法，但是其在处理一定数量的大数据时候还是有一定优势的。首先要生成 5 个不同数据规模的文件，这个随机数函数不太会用，去百度了以下随机数的生成办法，才得以解决。然后根据书上的算法写成插入排序函数和 shell 排序函数，在主函数中进行调用。并且还需要用到前阶段时间计算机组成与设计课程中的 clock 函数来计算 cpu 的执行时间，在 exe 执行文件中生成。同时也应用了上一次实验新接触的文件的读写。

实验过程中遇到的最大的问题就是插入排序在执行 1M 数据的时候需要很长的时间，昨天晚上刚听说一位同学跑了 2 个小时还没出结果的。我的是等了 40 分钟左右才有结果。这个应该与电脑的 cpu 有关系...

通过本次实验，通过程序验证了不同排序算法的性能。当执行大数据文件的时候真的有很大差距，之间在 c++ 编程的时候只接触过冒泡排序，但实际上也是最不实用的一种算法。快排，归并排序，堆排序等等会更实用一些。本次实验内容不多，因此一天完成。

至此，本学期的所有实验也完成了，认认真真做好实验无论是对考试还是代码能力都是一种提升，在学习算法的过程中，感受到了算法的实用。感谢数据结构这门课，教会了我很多非常实用的知识。也谢谢老师和助教的严谨和负责任。感谢！