实验一 排序算法

实验要求

使用各种排序算法对不同规模的字符串组和数组排序。用适当的方法或工具记录排序算法在执行时所消耗的时间。根据不同输入规模时记录的数据，画出算法在不同输入规模下的运行时间曲线图，并作分析。

实验环境

编译环境：Dev C++ 5.11，TDM-GCC 4.9.2 64-bit Debug

机器内存：16GB

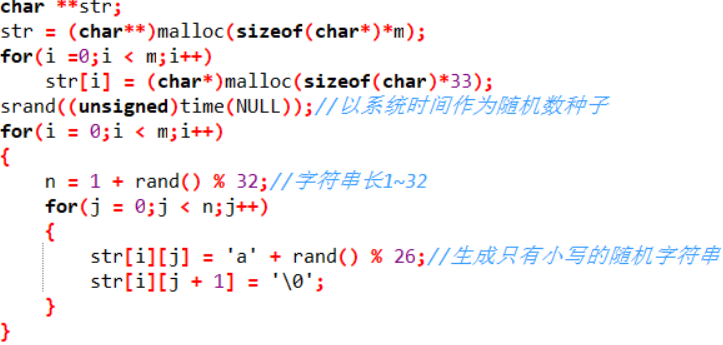
时钟频率：2.20GHz

实验过程

首先实现生成实验用随机字符串和数组的单独程序，然后分别用函数实现各排序算法，最后在主函数中调用排序算法，测量运行时间以及实现读写文件操作。

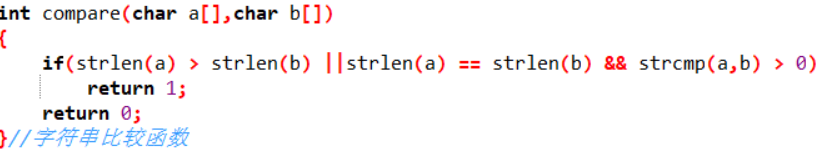
实验关键代码截图

1. 随机字符串（random-char）和数组生成(random-int)



以上为生成随机字符串的代码，由于直接定义2^17大小的二维字符数组会出错，所以最终改为使用二级指针动态管理。数组生成同理，不再赘述。

1. 字符串比较函数

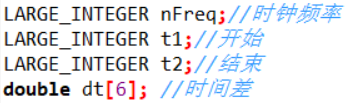


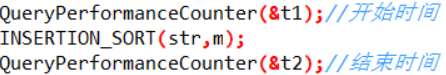
1. 各种排序函数

基本上由课本伪代码转化而来，只要注意函数传参和数组下标即可。

1. 测量算法运行时间

C:\Users\95850\AppData\Local\Microsoft\Windows\INetCache\Content.Word\5.png

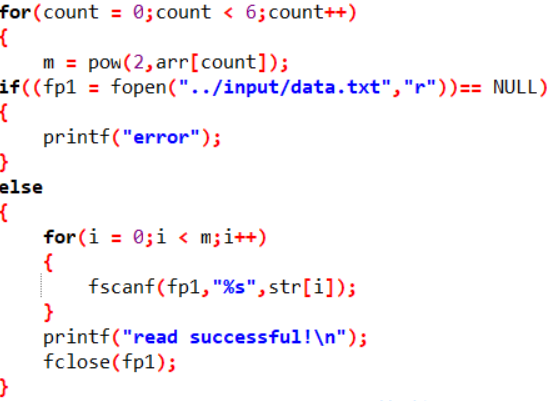


C:\Users\95850\AppData\Local\Microsoft\Windows\INetCache\Content.Word\7.png

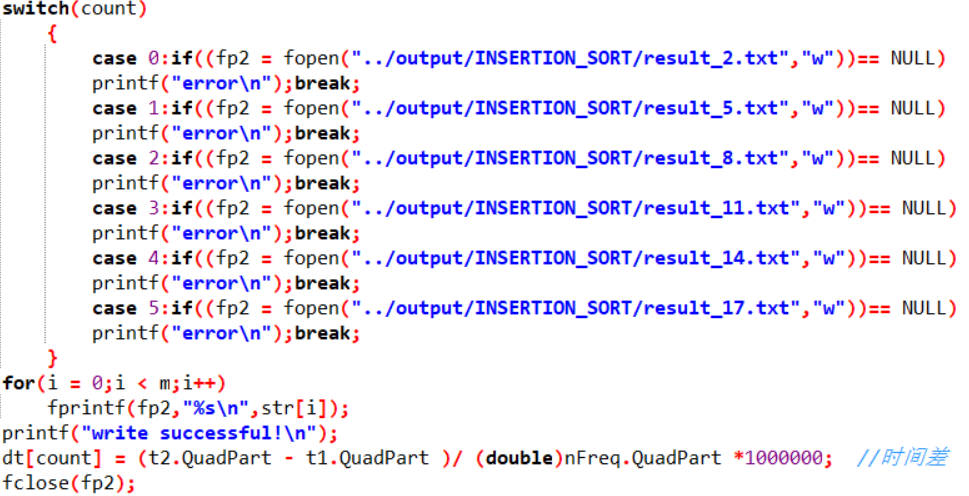
C:\Users\95850\AppData\Local\Microsoft\Windows\INetCache\Content.Word\9.png

1. 文件操作

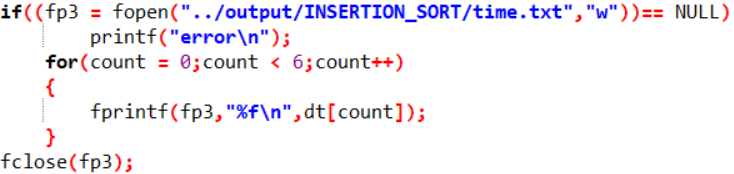
读待排序数据：



写排序后数据



写测量到的运行时间：



实验结果，分析

实验测得的运行时间整理在exp1表格和exp2表格中，运行时间图（纵坐标取lg2对数）如下：

可以看出，对于不同规模的数据，除了计数排序，其他排序算法运行时间均与数据规模呈正相关。实验一中快速排序，堆排序和归并排序运行时间的增长率相同，与课本中三者平均时间复杂度O（nlgn）一致；直接插入排序为O（n^2）与课本一致。实验二中冒泡排序为O（n^2），在2^11规模以后快排运行时间超过基数排序。计数排序基本保持O（n）的时间复杂度，但在数规模较小时，用时反而更大，这与计数排序需要额外的辅助数组有关。

对于相同规模不同排序算法的比较，实验一可以看出在2^2规模，插入排序运行最快，归并排序最慢，但在规模2^8以后，快排最快，堆排序和归并排序其次，插入排序非常慢。实验二可以看出规模较小时计数和基数排序很慢，这与这两个算法需要额外的辅助有关；在规模较大时，计数排序最快，冒泡排序最慢。

实验心得

本次实验让我复习了c语言的文件读写操作，认识到了c语言的局限性。通过对各种排序算法的分析，充分理解了各算法的基本原理，认识到了各算法的优缺点以及适用条件，对代码的优化有了一定的认识。学习了怎样测量程序运行时间以及图表的制作。