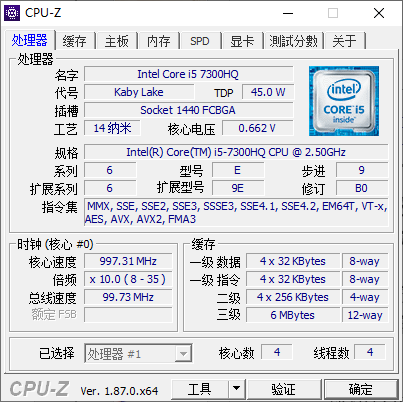
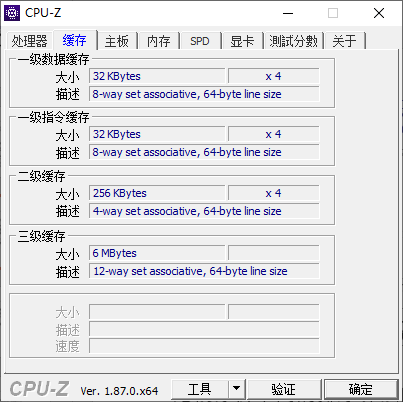
实验报告

1. 实验内容及要求

实验内容：排序n个元素，元素为随机生成的1到65535之间的整数，n的取值为： 23，25，27，29，211， 213 ，215 ；算法：堆排序，快速排序， 归并排序，计数排序。

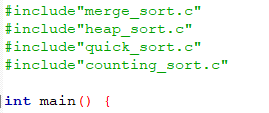
1. 实验设备和环境

笔记本使用两年四个月

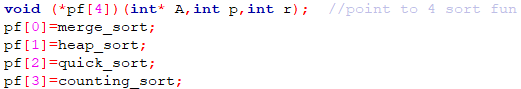


1. 实验方法和步骤

将main()单独写在一个.c里，四个排序算法分别单独写在一个.c里，方便调试。



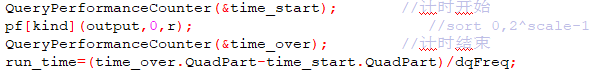
使用函数指针数组，分别指向四个排序算法，方便循环



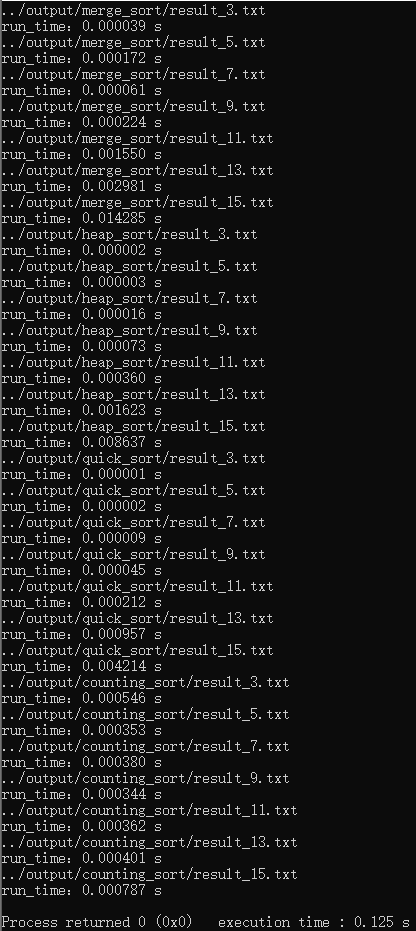
输出文件名根据循环信息拼接得到，方便循环



使用QueryPerformance计时，也就是



四个排序算法的实现参考书本，只是将用到动态数组的部分用malloc实现，输出如下



1. 实验结果与分析

将算法在不同输入规模下的运行时间曲线图画出，与期望相去甚远。

初步估计是因为比起算法真正执行的时间，call，return，cache调度占据的时间的更多，导致因数据量提升而带来的运行时间提升不明显，（甚至还有下降，T\_T）

因此，在数据规模较小的时候，运行时间随数据规模的变化看不懂，但在数据规模大了后由于样本太少也看不出来n\*log（n）的曲线。

虽然数据不咋好看，但在不同数据规模下哪一种算法更占优还是看得出一些的

merge\_sort在本次实验中各方面表现都不行

heap\_sort比上一个好很多，曲线也是正常递增，但各方面都被quick\_sort完爆

counting\_sort在数据规模小时由于C数组也要开到65535导致运行时间较长，但在数据规模>2^11时表现良好