算法导论实验报告

PB14209127 范子健

注：请在 linux 下查看代码，如在 windows 下，请修改文本格式为 utf-8,不然，部分中文注释可能会变成乱码。另外，由于 linux 和 Windows 下换行方式不一样，在 Windows 下打开 time.txt 文件时，文本不会换行。

1. 实验要求：

a) 实验一：排序 n 个元素，元素为随机生成的长为 1..32 的字符串（字符串均为英文小写字母），n 的取值为：2^2，2^5，2^8，2^{11}，2^{14}，2^{17}；算法：直接插入排序，堆排序，归并排序，快速排序。
b) 实验二：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： 2^2，2^5，2^8，2^{11}，2^{14}，2^{17} ；算法：冒泡排序，快排，基数排序，计数排序
c) 字符串大小比较：
    a) 首先按照字符串长度进行排序，短的在前面，长的在后面，
    b) 若长度相同，则按照首个不同的字符顺序排序

2. 实验环境：

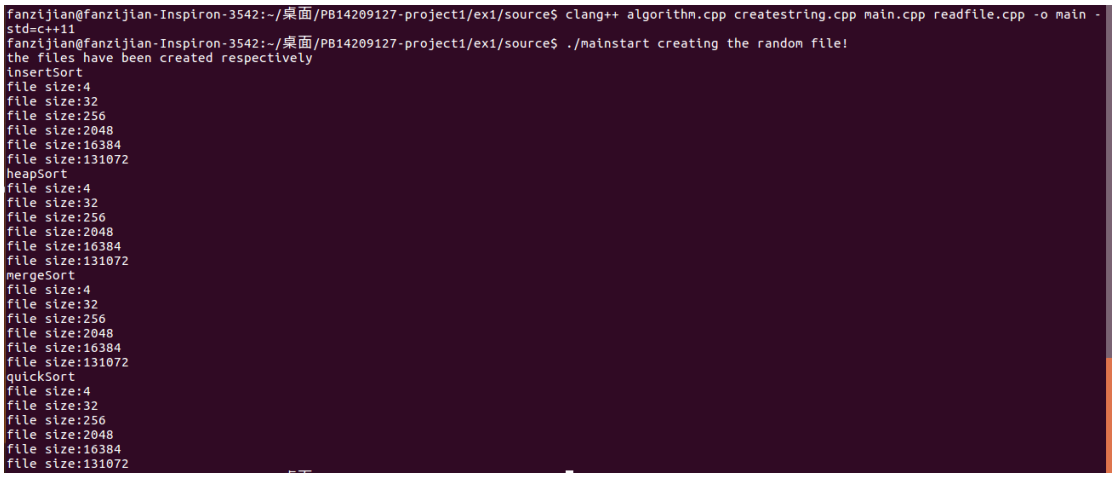
编译环境：Clang 5.0.0，ubuntu 16.04 release
机器内存：4GB
CPU 频率：1.70GHZ

3. 实验过程：

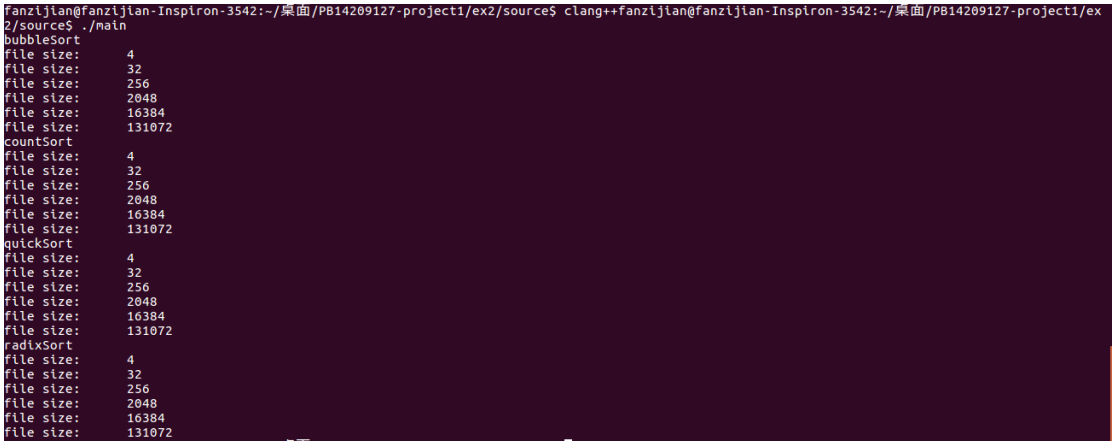
3-1. 实验设计：

a) 我是先实现 ex2,然后再实现 ex1 的，所以，ex2 代码的结构相比 ex1 显得混乱些。具体的不同会在稍后阐述。
   实验的代码分成三个模块，分别为随机数据生成，文件读取，算法执行并写入结果，在随机数据部分，使用的是 srand()和 rand()函数，以生成大致符合正态分布的伪随机数。在文件读取部分，使用 C++的标准库 fstream,算法部分和课本相同，不作赘述。
b) 源代码已经按照要求放在相应的文件夹中，运行(比如运行 ex1)时先 cd 进 PB14209127-project/ex1/source/中，然后打开终端，输入命令行
    i. clang++ algorithm.cpp createstring.cpp main.cpp readfile.cpp -o main -std=c++11
    ii. 然后运行 ./main，输出如下所示。

iii.



iv.

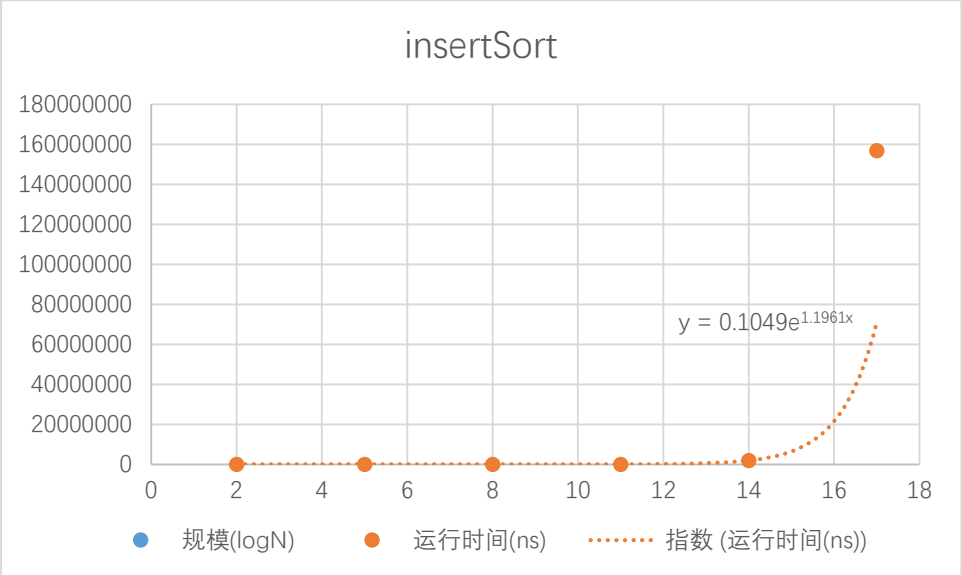


v. 和 ex2 相比，ex1 每次都会重新生成随机数据，而 ex2 则是在项目之外，用另外一个程序生成随机数，助教可以在 ex2/input/createRandom/文件夹下找到相应的源码。

4. 实验结果以及分析：

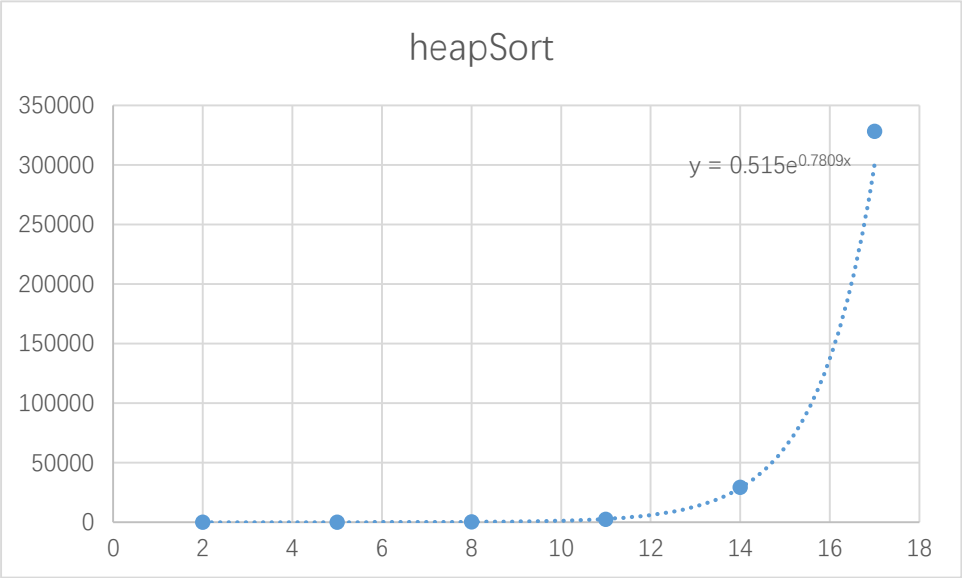
Ex1:

Insert sort 性能曲线：



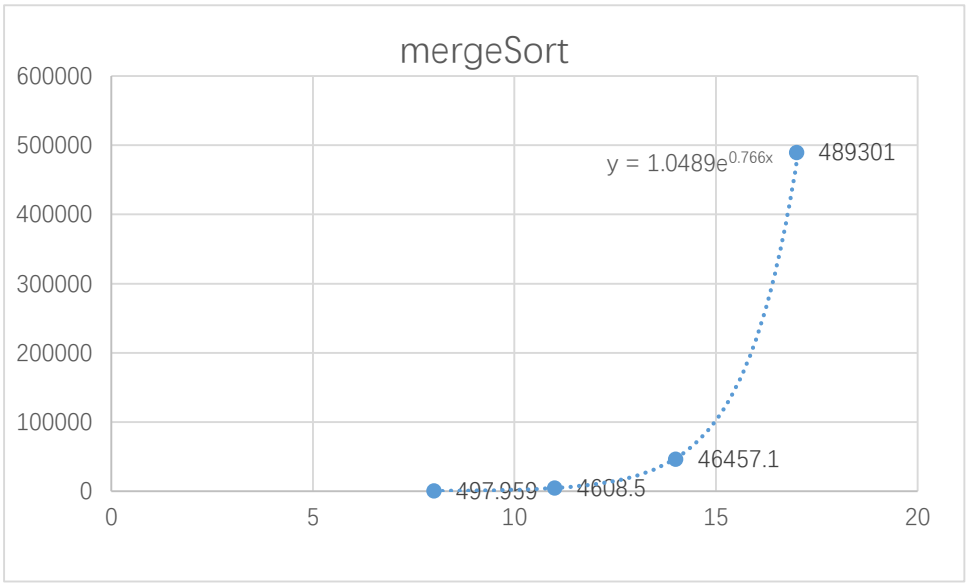
横坐标为数据规模，单位为 log(N),纵坐标为运行时间，单位为 ns (之后的图表单位一样)，由图可知 ，曲线大致符合指数增长，而且指数相比其他排序算法要大，因为我的横坐标取得是规模的对数，所以与教材上的结论是吻合的。

Heapsort:



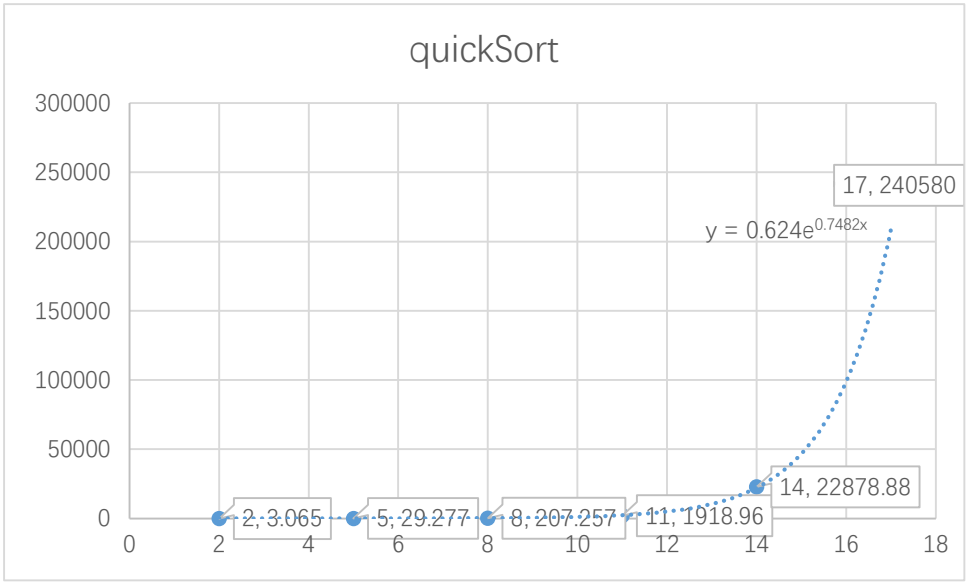
坐标轴单位与意义与上述类似，符合教材结论，可以看到，heapSort 指数明显小于 insertSort 的指数，

mergeSort:



实验结果与教材结论吻合，性能与 heapSort 相近。

QuickSort:

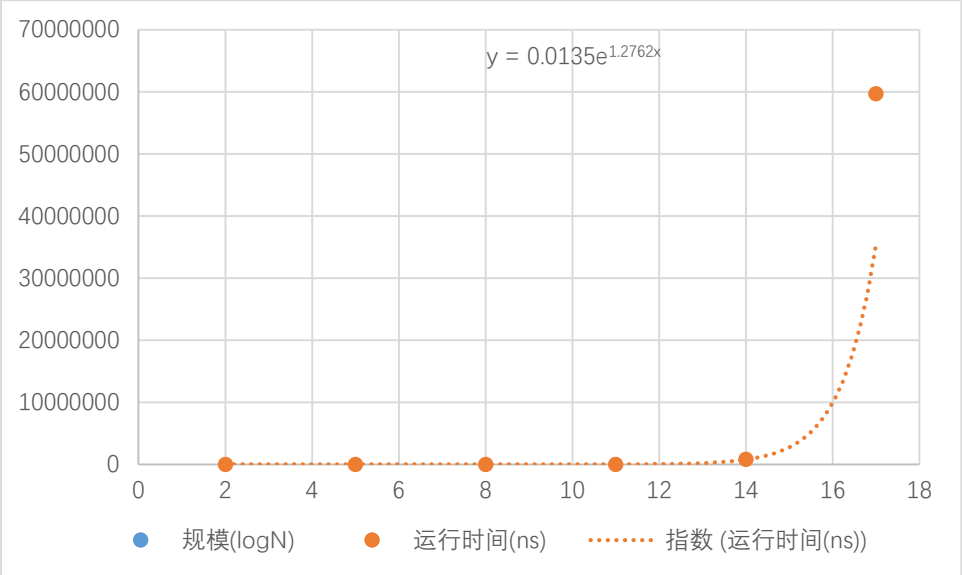


符合教材所给的结论。

综合比较上述四个排序方法，发现性能最好的是快排，各个实验结果的到拟合函数的指数按照从大到小的顺序分别为：1.196，0.7809，0.766，0.7482，可以发现 insertSort 的指数是最大的，其他三种排序方法差别不大。

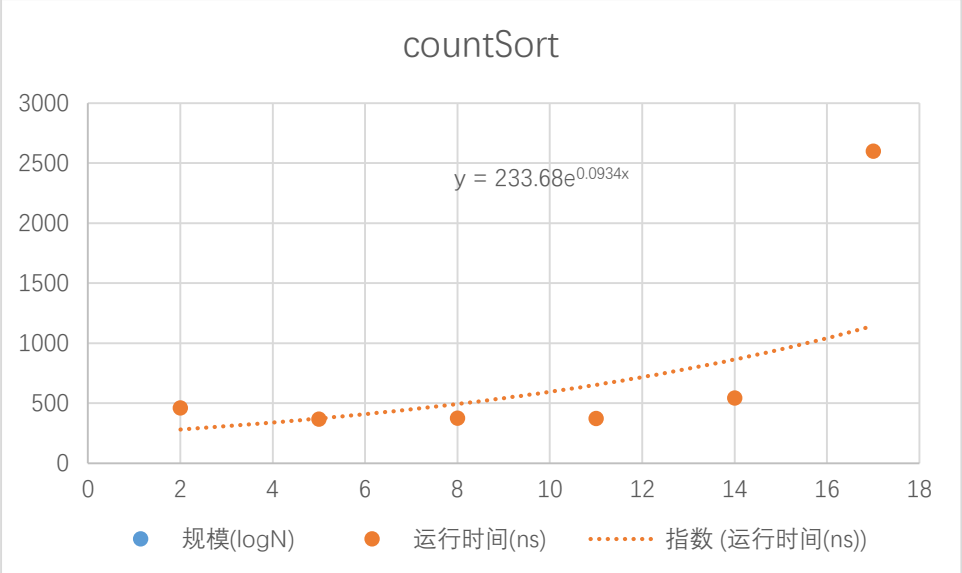
Ex2:

bubbleSort:



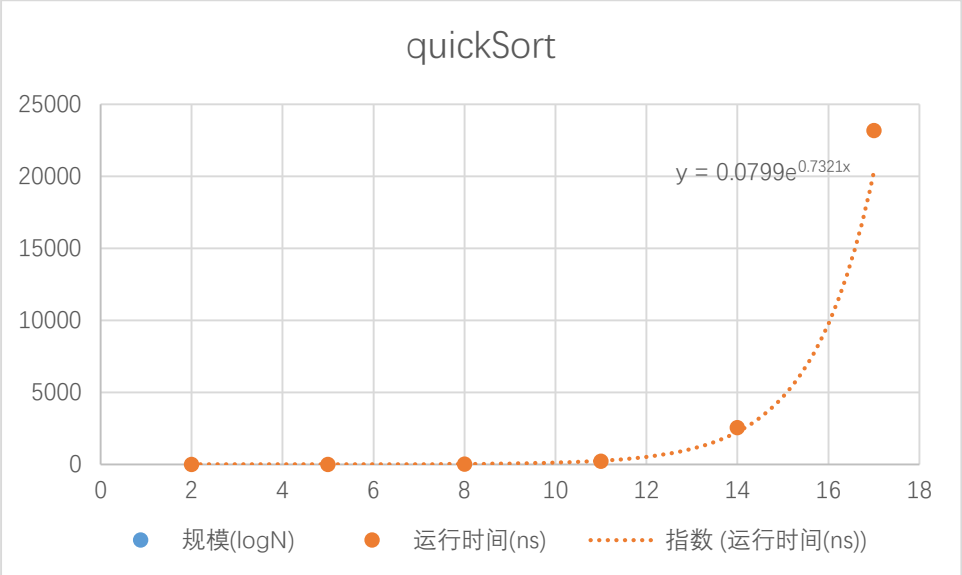
和结论相符，符合指数函数

CountSort:



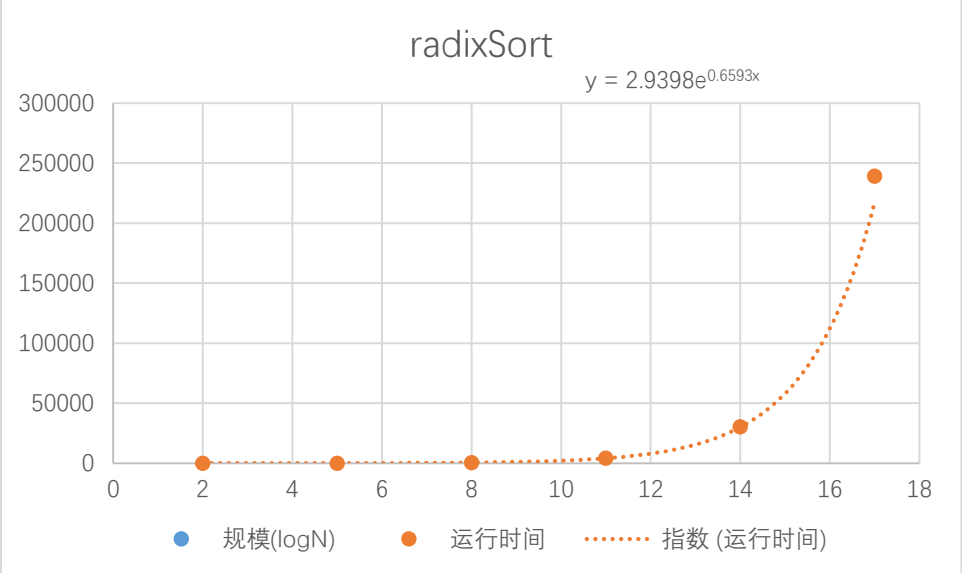
countSort 的曲线与指数曲线大致符合，但是不太好，我也尝试了其他拟合函数，但是都不如指数函数好。按照课本上的结论，时间复杂度应当为 O(N),大致上是符合的。

Quicksort：



与结论符合得很好，

radixSort：



与结论符合得很好。

比较上述四种算法，bubble sort 最差，quick sort 最好，在数据规模较大时，bubblesort 性能远远落后于其他三种算法，count sort 算法则相对平稳一些，预计随着数据规模增大（数据范围不变），count sort 性能会超过快排。

5. 实验总结：

在实现算法的过程中，遇到了一些问题，特别是边界检查，访问越界，一度耗费大部分时间来处理这些问题，但是也加深了对算法的理解，这一点，仅仅看书是无法达成的，以后还是得多多实践。