实验报告

目录

实验环境

硬件环境

电脑：实验室台式机
CPU：Intel 酷睿 i7 8700
内存：64GB
硬盘：1TB SSD

实验时无其他程序干扰，只开了 IDEA，硬盘容量和内存容量很充足

软件环境

操作系统：Windows 10 1903 家庭中文版
IDE：IDEA
编程语言：JAVA
JDK 版本：JDK 13。下载来源：Oracle 官网
编译器：javac 13
编译选项："C:\Program Files\Java\jdk-13\bin\java.exe" -Xmx32768m --enable-preview "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019.2.3\lib\idea_rt.jar=50867:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019.2.3\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\QT\Desktop\算法\第二次作业排序\out\production\第二次作业排序 Main

实验结果

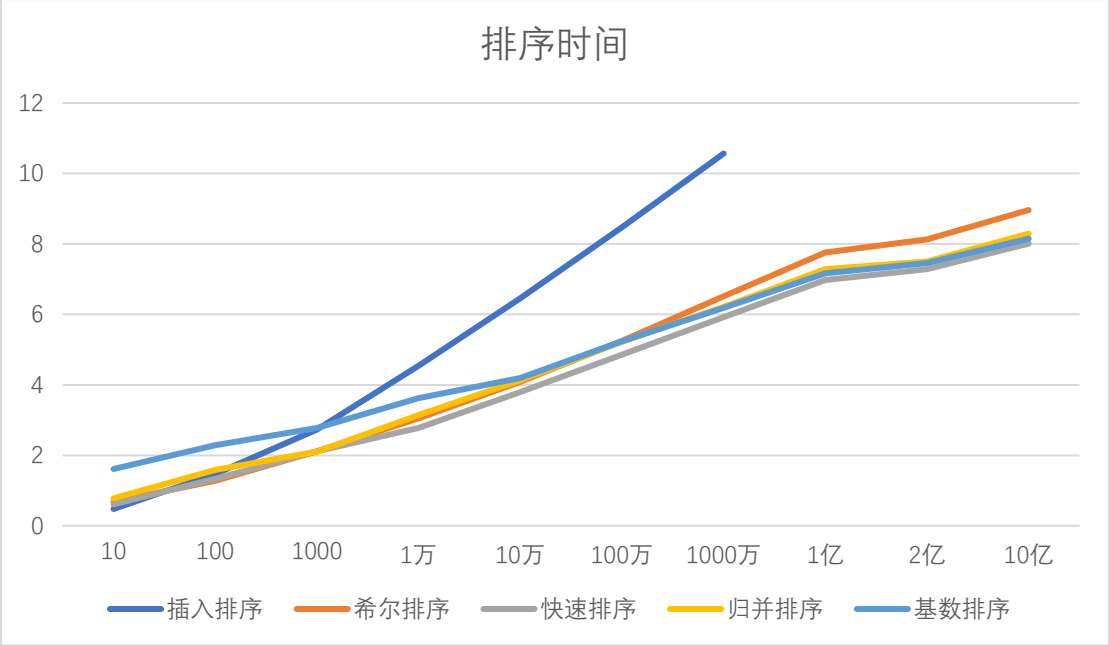
实验过程：每种排序，每个数量级，跑 3 次，取平均值。插入排序在 1 亿及以上数量级没跑。

	插入排序	希尔排序	快速排序	归并排序	基数排序
10	4.49us	7.38us	6.74us	10.27us	37.53us
	2.25us	3.53us	2.57us	3.85us	59.67us
	2.25us	3.53us	2.89us	3.85us	25.66us
100	64.80us	38.17us	20.85us	38.5us	191.2us
	11.87us	9.30us	21.17us	42.02us	194.73us
	12.19us	9.30us	22.78us	34.65us	191.84us
1000	1.01ms	124.15us	243.49us	123.19us	883.17us
	335.88us	171.63us	73.78us	119.98us	920.70us
	286.80us	91.11us	71.86us	141.79us	1.38ms
1 万	27.97ms	992.88us	542.16us	1.22ms	7.90ms
	28.55ms	977.48us	536.38us	1.23ms	2.61ms
	48.24ms	1.42ms	705.12us	1.73ms	2.21ms
10 万	2.84s	12.78ms	6.66ms	14.94ms	16.93ms
	2.87s	12.18ms	6.19ms	13.38ms	15.00ms
	2.86s	11.88ms	6.03ms	12.55ms	15.35ms
100 万	284.60s	175.35ms	72.21ms	167.80ms	168.35ms
	311.68s	177.96ms	72.54ms	168.20ms	182.70ms
	299.82s	175.55ms	72.78ms	168.45ms	170.44ms
1000 万	36916.88s	3.24s	832.65ms	1.66s	1.52s
	36907.37s	3.30s	846.41ms	1.65s	1.52s
	36865.31s	3.34s	842.92ms	1.59s	1.51s
1 亿	无	57.87s	9.73s	19.08s	15.19s
	无	57.26s	9.27s	19.36s	14.74s
	无	54.98s	9.55s	19.15s	14.54s
2 亿	无	136.96s	19.61s	19.61s	28.91s
	无	132.34s	19.20s	38.00s	28.82s
	无	137.08s	18.91s	37.57s	28.32s
10 亿	无	913.65s	102.94s	197.78s	141.37s
	无	920.71s	101.90s	195.67s	141.51s
	无	925.24s	102.73s	195.19s	141.35s

以上是原始数据，以下是平均时间表（因为都塞在一张表里，word 太挤了，一页就那么宽）

	插入排序	希尔排序	快速排序	归并排序	基数排序
10	3us	4.81us	4.07us	5.99us	40.95us
100	29.62us	18.92us	21.6us	38.39us	192.59us
1000	544.23us	128.96us	129.71us	128.32us	601.75us
1 万	34.92ms	1.13ms	594.55us	1.39ms	4.24ms
10 万	2.86s	12.28ms	6.29ms	13.62ms	15.76ms
100 万	298.7s	176.29ms	72.51ms	168.15ms	173.83ms
1000 万	36896.52s	3.29s	840.66ms	1.63s	1.52s
1 亿	无	56.70s	9.52s	19.2s	14.82s
2 亿	无	135.46s	19.24s	31.73s	28.68s
10 亿	无	919.87s	102.52s	196.21s	141.41s

下面以平均值表格的数据画图，所有的数据以 us 为单位，取 10 为底的底数，由于 1 亿及以上插入排序跑不出数据，可认为时间是无穷大



实验分析

1. 插入排序在数据量为 10 的时候时间最少，后面就不行了，增长太快。
2. 基数排序一开始比较慢，随着数据量增大比除了快排之外的算法渐渐快了。
3. 在后期的竞争中，希尔排序渐渐败下阵来。归并排序，快速排序和基数排序不相上下。
4. 快速排序名副其实，在各个数据规模下都很强。
5. 实验结果自认为很完美。

小结

遇到的问题

1. 最开始在自己的笔记本上跑的，只有 8G 内存，操作系统启动后剩 5G，跑到后面发现内存不够，然后把程序拷到实验室 64G 内存的台式机上跑。
2. 插入排序在 1 个亿数据量的时候，跑了 1 天+还没跑完，果断放弃，把插入排序丢掉了继续跑。所以 1 个亿及以上的数据中，没有插入排序。

改进空间

1. 一趟希尔我是用直接插入排序做的，如果用二分插排还能更快，理论上还有进步的空间
2. 因为图是用以 10 为底的对数画的，虽然 y 坐标归一化了，但是各种排序之间的差异也模糊了，实际上，快排的时间几乎稳定在其它 nlogn 算法的一半，但是从图上看不出来。