

多种排序方法实验报告

洪方舟

Student ID: 2016013259

Email: hongfz16@163.com

March 17, 2018

1. 实验目的

- a. 编写程序实现插入排序，希尔排序，快速排序，归并排序，基数排序
- b. 输入数据规模为 $10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 10^8, 2 \times 10^8, 10^9$ 的数组，记录不同排序方式所使用的时间
- c. 加深对于各种排序方式优缺点的认识
- d. 编写生成32位无符号整数随机数的程序

2. 实验环境

操作系统：ubuntu 16.04 LTS

处理器：Intel Core i7-7700k CPU @ 4.20GHz * 8

编程语言：C++

编译器：g++

3. 实验方法

- 编写五种排序方法，通过遍历判断的方法保证排序的正确性
- 编写生成32位无符号整数的随机数程序方法：通过实验可知，实验环境中随机数生成函数 $rand()$ 可以生成的最大数为 $2^{31}-1$ ，将该数左移一位，并且用随机数生成函数生成的数模2 ($rand()\%2$)，来填充最低位，这样生成的32位无符号整数随机数在 $0 - 2^{32} - 1$ 的范围内是等概率的
- 通过 $ctime$ 的库函数 $clock()$ 来实现计时功能

4. 实验结果

Table 1: 不同排序方法在不同数量级数据下所耗时间(ms)

数据大小	基数排序	快速排序	归并排序	希尔排序	插入排序
10	0.043	0.003	0.002	0.003	0.002
10^2	0.059	0.025	0.024	0.037	0.036
10^3	0.338	0.341	0.313	0.106	0.489
10^4	0.57	0.838	0.916	7.058	48.57
10^5	5.89	11.183	12.506	541.713	4722.16
10^6	60.738	131.824	142.881	55379.6	×
10^7	604.196	1593.94	1691.85	×	×
10^8	6027.44	17864.9	19856.9	×	×
2×10^8	12908.4	37015.5	41291.3	×	×
10^9	60971.5	199088	225937	×	×

5. 分析与总结

- 通过验证，程序具有一定的正确性
- 由实验结果可以验证，各算法的时间复杂度基本符合理论结果
- 在较大数据量的排序中，基数排序具有较大的时间复杂度优势，但同时所使用的空间限制了基数排序在更大数据量的数据排序中的应用
- 在 $(n \lg n)$ 的两个算法中，快速排序明显快于归并排序，这与预期结果一致

e. 希尔排序和插入排序渐进时间复杂度的增长过快，因此没有进行较大数据量的实验，但是两者在小数据规模的表现较好，因此可以将两者作为递归排序算法递归终止后的小数组排序算法

6. 源代码及可执行文件说明

a. 源代码存放在 */src/SortSrc*

b. 可执行文件存放在 */bin/Sort*

c. 点击可执行文件之后会自动开始测试各排序算法，并给出所耗时间，大数量级的数据测试中自动排除了希尔排序和插入排序，仅测试另外三种以节约时间