

算法实验一

Pb18081616 谭园

一、实验要求

- 1 排序 n 个元素, 元素为随机生成的 0 到 $2^{15}-1$ 之间的整数, n 的取值为: $2^3, 2^6, 2^9, 2^{12}, 2^{15}, 2^{18}$ 。
- 2 实现以下算法: 直接插入排序, 堆排序, 快速排序, 归并排序, 计数排序

二、实验环境

编译环境: DEV C++

机器内存: 16GB

时钟主频: 2.3GHz

三、实验过程

1. 利用 rand 函数生成了 300000 个随机数(1-32768), 并写入 input 文件中。
- 2 利用 windows.h 附带的函数得到程序运行时间。us 级别。
3. 写出各排序算法的实现并进行排序
4. 得出结果并分析。

四、实验结果:单位(us)

规模\算法	插入排序	堆排序	快速排序	归并排序	计数排序
3	0.2	0.6	0.5	3.8	227.1
6	4.4	6.1	5	12.9	193.1
9	206.6	57.6	48.6	237.6	219.9
12	10964.3	589.3	374.8	14197.4	232.9
15	606832.3	5887.5	3576.4	787568.3	473.8
18	38767247.9	59198.4	33267.1	50479820.8	3261.2

N=2³时的排序截图:

CT_result_3.txt - 记事本

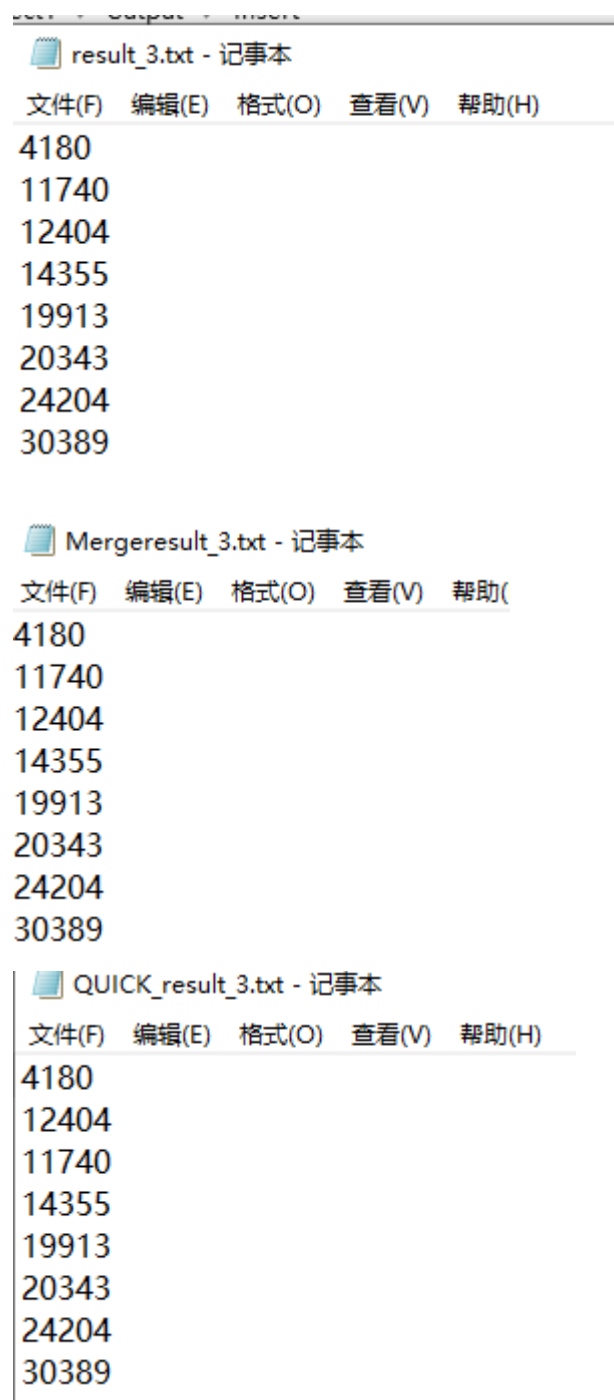
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

4180
11740
12404
14355
19913
20343
24204
30389

Heap_result_3.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

4180
11740
12404
14355
19913
20343
24204
30389



运行时间截图：

规模为 3：

```
C:\Users\15564\Desktop\2020算法\project1.exe

插入排序时间是: 0.200000us
堆插入排序时间是: 0.600000us
快速排序时间是: 0.500000us
归并排序时间是: 3.800000us
计数排序时间是: 227.100000us
-----
Process exited after 0.01739 seconds with return value 1
请按任意键继续. . .
```

```
C:\Users\15564\Desktop\2020算法\project1.exe

插入排序时间是: 4.400000us
堆插入排序时间是: 6.100000us
快速排序时间是: 5.000000us
归并排序时间是: 12.900000us
计数排序时间是: 193.100000us
-----
Process exited after 0.01921 seconds with return value 1
请按任意键继续. . .
```

```
C:\Users\15564\Desktop\2020算法\project1.exe
插入排序时间是: 206.600000us
堆插入排序时间是: 57.600000us
快速排序时间是: 43.600000us
归并排序时间是: 237.600000us
计数排序时间是: 219.900000us
-----
Process exited after 0.01638 seconds with return value 1
请按任意键继续. . .
```

```
C:\Users\15564\Desktop\2020算法\project1.exe
插入排序时间是: 10964.300000us
堆插入排序时间是: 589.300000us
快速排序时间是: 374.800000us
归并排序时间是: 14197.400000us
计数排序时间是: 232.900000us
-----
Process exited after 0.04413 seconds with return value 1
请按任意键继续. . .
```

```
C:\Users\15564\Desktop\2020算法\project1.exe

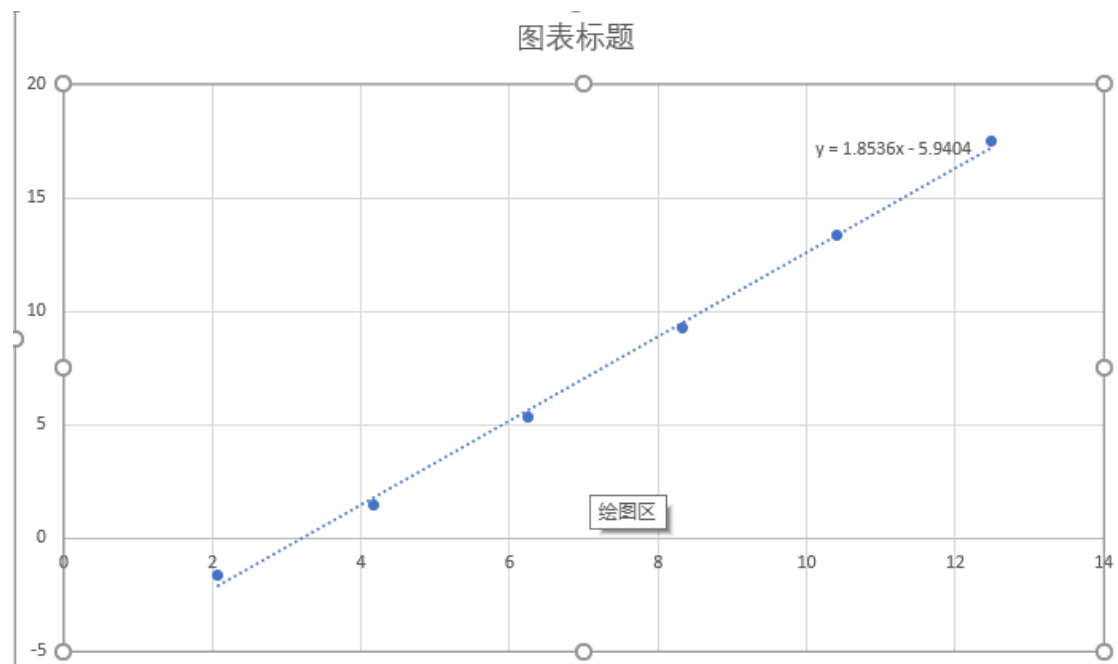
插入排序时间是: 606832.300000us
堆插入排序时间是: 5887.500000us
快速排序时间是: 3576.400000us
归并排序时间是: 787568.300000us
计数排序时间是: 473.800000us
-----
Process exited after 1.453 seconds with return value 1
请按任意键继续. . .
```

```
C:\Users\15564\Desktop\2020算法\project1.exe

插入排序时间是: 38767247.900000us
堆插入排序时间是: 59198.400000us
快速排序时间是: 33267.100000us
归并排序时间是: 50479820.800000us
计数排序时间是: 3261.200000us
-----
Process exited after 89.58 seconds with return value 1
请按任意键继续. . .
```

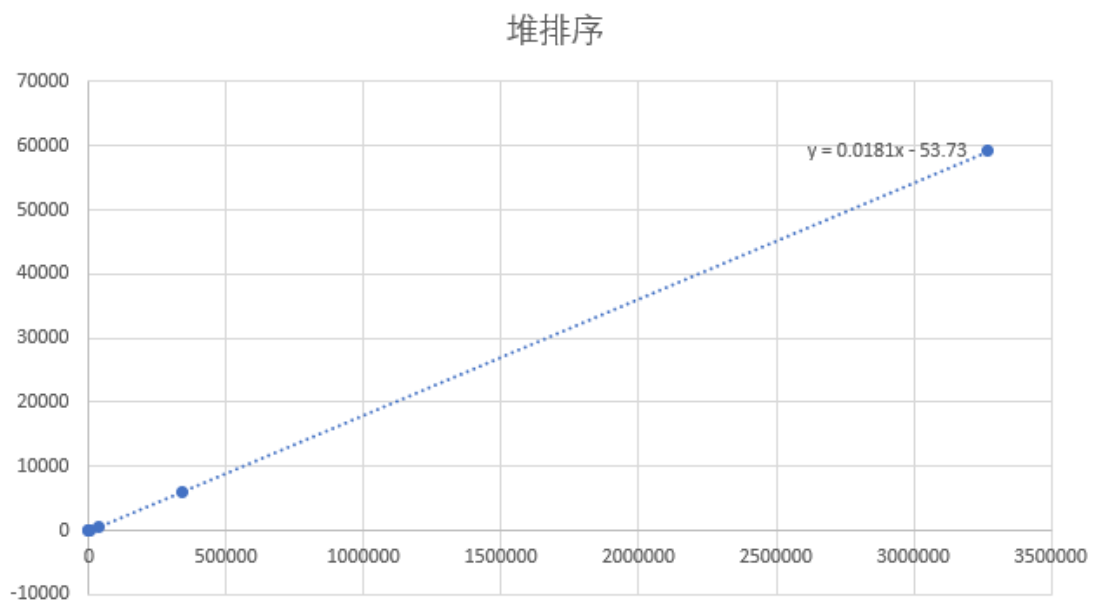
五、实验结果分析

1. 插入排序, 横纵坐标都取对数再拟合



可以看出, 斜率很接近 2, 即说明这个的插入排序算法的时间复杂度是满足 $\theta(n^2)$ 的。

2.堆排序

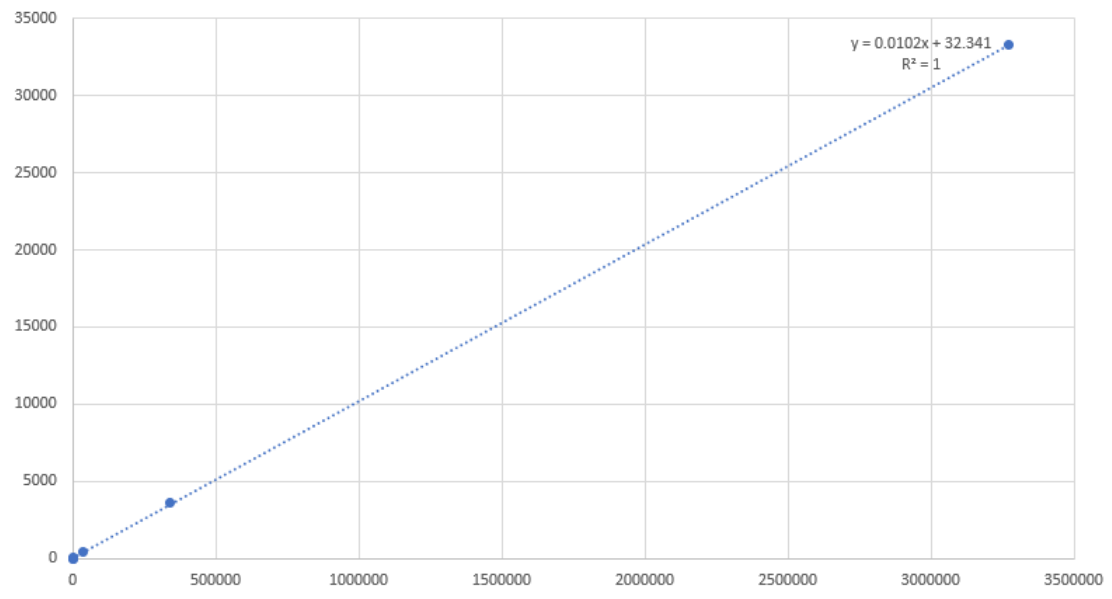


以 $n \cdot \ln n$ 为自变量, 最后拟合直线效果很好, r 的平方值为 1, 可以证明算法复杂度满足 $y = \theta(n \cdot \ln n)$

3.快速排序

和堆排序一样的复杂度, 拟合方式也一样。

快速排序

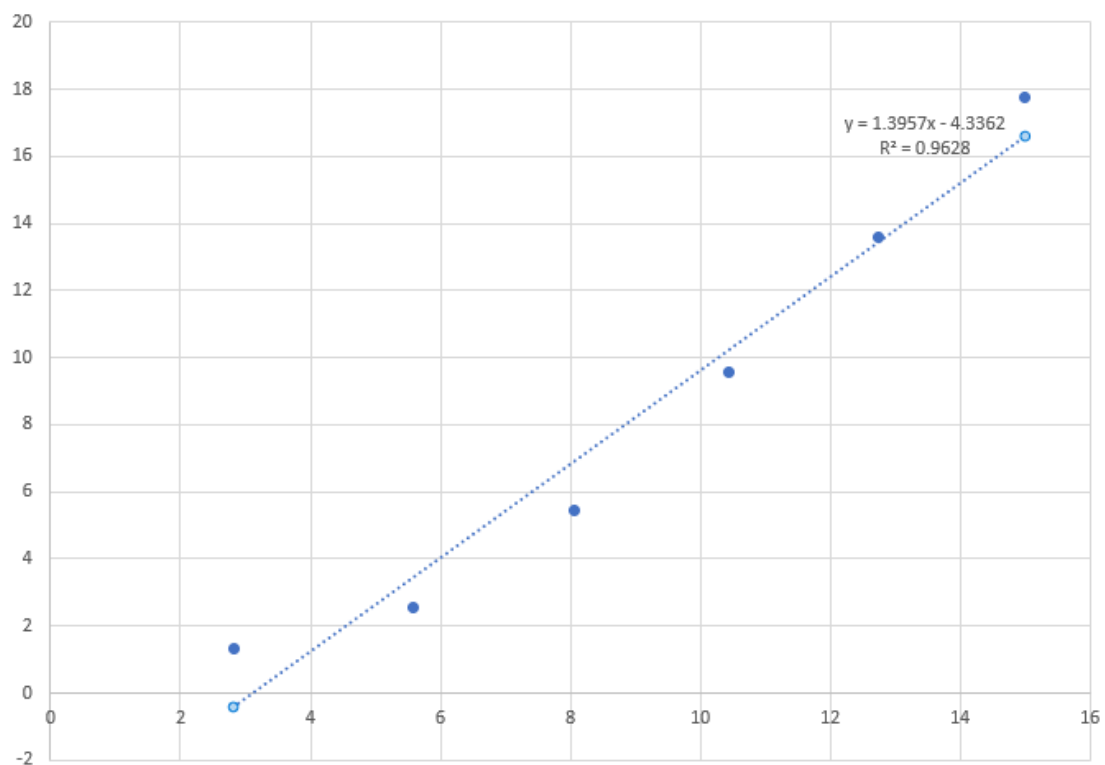


以证明算法复杂度满足 $y = \theta(n \cdot \ln n)$

4. 归并排序

归并排序的算法复杂的同上，将横纵坐标取对数后得到下图

归并排序

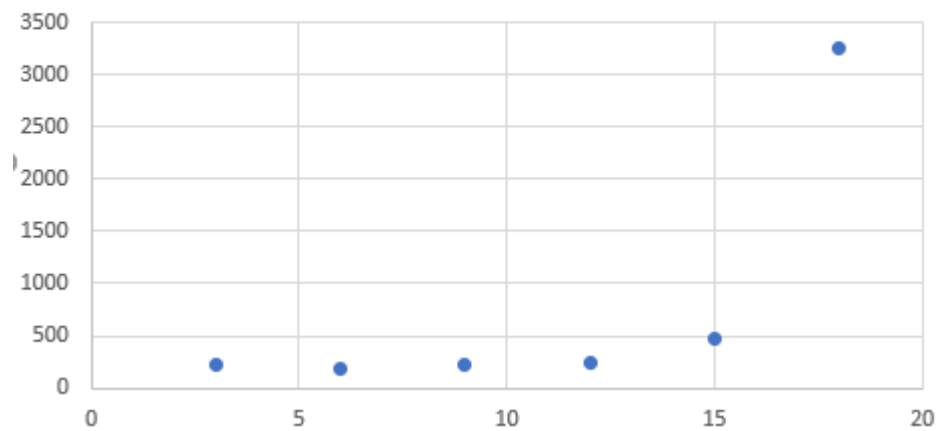


拟合的不是很好，应该是我算法实现的比较不稳定，导致时间波动有点大。

5. 计数排序

将 x 取对数后作图，

计数排序



计数排序的时间复杂度是 $\theta(k+n)$ ，此处 k 的值是 32767 即 $2^{15}-1$ ，而前四个数据的规模分别是： $2^3, 2^6, 2^9, 2^{12}$ ，此时 $n \leq k$ ，所以前四次排序消耗的时间相差无几。第 5 组数据规模是 2^{15} ，可以看到，排序第五组数据消耗的时间约为前四组数据 2 倍，非常契合时间复杂度的分析。第六组数据的规模是 2^{18} 次方，此时 $n > k$ ，这个计数排序消耗的时间主要由数据的规模大小决定，并且明显大于前几组数据消耗的时间，这和时间复杂度的分析和渐进曲线都很契合

得出结论：

仅从数据看出，课本上对各种排序方法的分析和实际情况都是基本契合的。总的来说，实验结果是令人满意的。

比较：

从实验结果我们可以得到以下结论：

- 1、插入排序的时间复杂度是最大的，通常情况下，我们都应该避免采用插入排序。
- 2、归并排序，堆排序和快速排序：通常情况下，快速排序的表现都比堆排序要好。根据本次实验的数据，我们可以看到，虽然时间复杂度都是 $n \cdot \ln n$ 量级，但是快速排序在排序我们随机生成的随机数是消耗的时间大约都是堆排序的 $1/2$ ，并且快速排序是原地排序，堆排序需要额外的空间。但是快速排序的最坏情况非常糟糕。
- 3、计数排序：在数据规模 n 比数据范围 k 大很多并且存储空间足够时，应该尽量采用计数排序，计数排序会消耗不少的额外空间，但是在数据规模很大时，它的表现比快速排序和堆排序还要好。

综上所述，在数据规模很大时，计数排序的优势是最大的，但是快速排序，归并排序，堆排序的时间消耗也是可以忍受的，并且空间复杂度比计数排序小。在实际应用中应该根据情况进行取舍，插入排序算法简单，在数据规模很小时也是可以用的。

六、实验心得

学完排序算法以后，我的知识还停留在书本上，对各种排序算法只能说知道，通过这次实验，自己实现各种排序算法，得到真实的数据，更加深入地理解了这几个排序算法。通过自己的实践，验证了课本上关于排序算法的时间复杂度分析，更深刻认识到了时间复杂度分析的重要性，它能帮助我们省下许多宝贵的时间，重要性不亚于算法设计本身。

对不同规模的数据排序，各有不同的最优算法，这启发我们要不能死板，要结合实际情况来采用合适的算法完成工作。

