

# 算法基础上机实验一

PB17000285

雷洋

## 一．实验内容

1. 对  $n$  个元素进行排序，元素为随机生成的 1 到 65535 之间的整数， $n$  取值为  $2^3$ ,  $2^5$ ,  $2^7$ ,  $2^9$ ,  $2^{11}$ ,  $2^{13}$ ,  $2^{15}$ 。
2. 采用算法：堆排序，快速排序，归并排序，计数排序。

## 二．实验要求

1. 实验文件严格按照要求创建。
2. 输入文件每行一个随机数据，总行数大于等于  $2^{15} = 32768$  顺序读入指定规模。
3. 每种算法建立子文件夹，result\_n.txt 对应  $2^n$  规模的输出，time.txt 对应七个规模的时间输出。

## 三．实验设备及环境

实验设备和环境为 windows10 个人计算机操作系统下的 devC++ IDE, TDM-GCC 4.9.2 64-bit。

## 四．实验方法

1. 通过伪随机函数 rand () 的变形来生成 0-65536 范围的随机数，共 65536 行，文件写入 input 文件夹的文件中。
2. 通过文件读写来输入数据，存放在临时动态申请的整型数组中，在不同的算法中传入数组头指针，开始位置 1 和结束位置  $2^n$ ，（在计数排序中还需要传入输出数组 B 的头指针，不过

不需要传入开始位置)。将排好序的数组写入 result 文件。

3. 通过 windows 库里的函数获取 cpu 频率，以及开始和结束的周期数差，来计算运行时间（单位秒），文件写入 time.txt 中。这里统一对每次排序的 7 个规模的时间定义格式：

time(t1,t2,t3,t4,t5,t6,t7),其中  $t_n$  由前到后分别是规模从最小到最大的时间，方便通过宏定义快速创建数组。

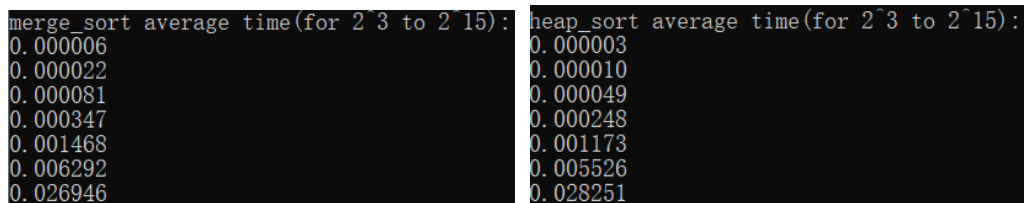
4.重复统计 50 次各个排序算法的各个规模时间，以其平均数来作为最后分析的数据，绘图观察分析得出结论。

## 五．实验步骤

1. 创建头文件 sort.h,写入必要的宏定义，声明和定义所用到全局变量和函数。
- 2.创建 create\_data.cpp，生成随机数表。
3. 分别创建四种算法的.cpp 文件。
4. 创建 main.cpp，对不同算法不同规模的排序时间进行统计和输出。
5. 创建 analysis.cpp，对 50 次排序时间的统计进行求平均数。

## 六．实验结果与分析

1. 本次实验结果截图：



```
merge_sort average time(for 2^3 to 2^15):
0.000006
0.000022
0.000081
0.000347
0.001468
0.006292
0.026946

heap_sort average time(for 2^3 to 2^15):
0.000003
0.000010
0.000049
0.000248
0.001173
0.005526
0.028251
```

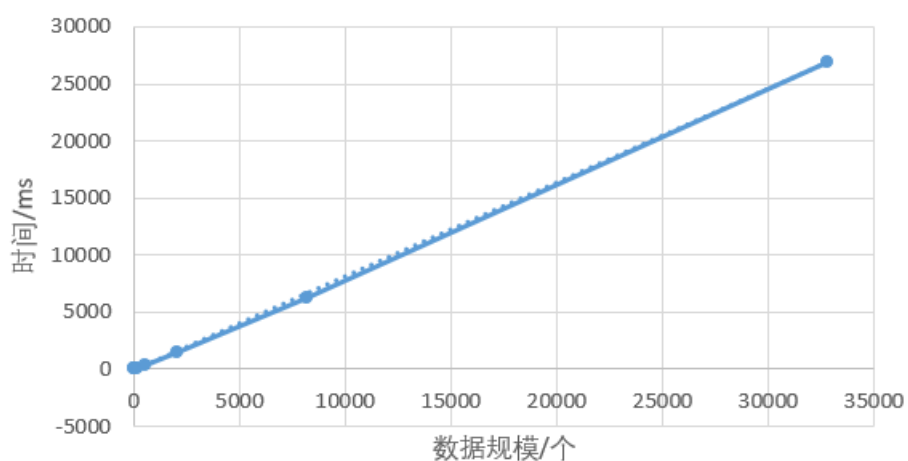
```
quick_sort average time(for 2 3 to 2 15):
0.000001
0.000005
0.000028
0.000136
0.000637
0.002922
0.014503
```

```
counting_sort average time(for 2 3 to 2 15):
0.001175
0.001218
0.001184
0.001207
0.001253
0.001440
0.002221
```

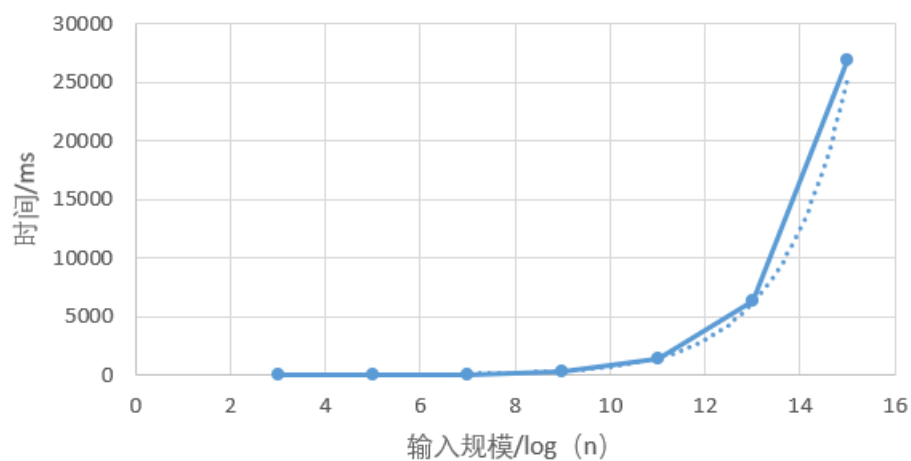
## 2.结果分析

### 1>归并排序:

归并排序时间趋势



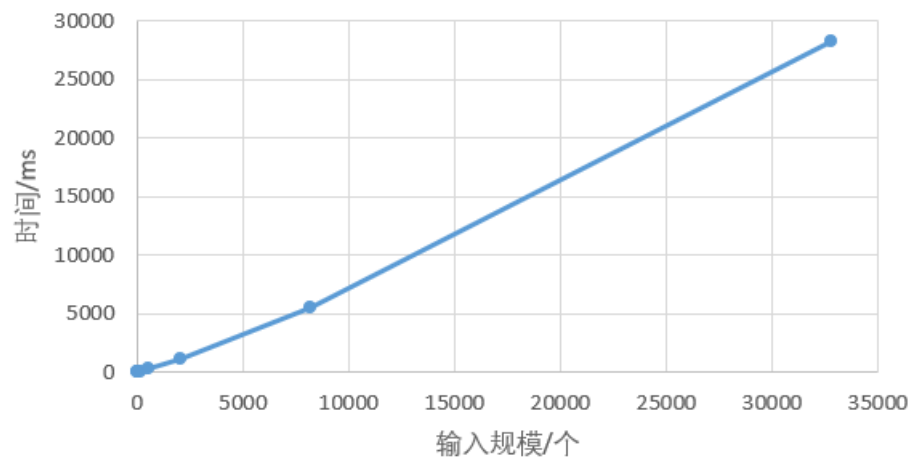
归并排序时间趋势



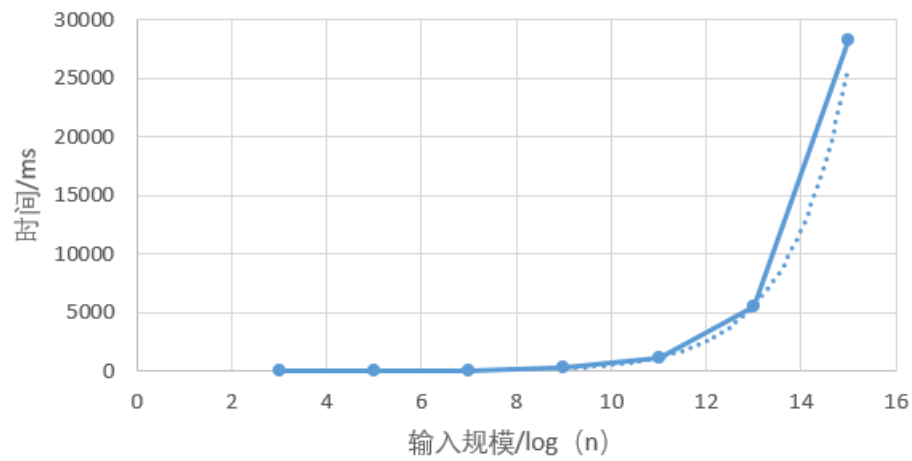
通过俩个图像的综合分析，可以看到归并排序是的渐进时间大致符合 $\Theta(n \log_2 n)$ 。

### 2>堆排序

堆排序时间趋势



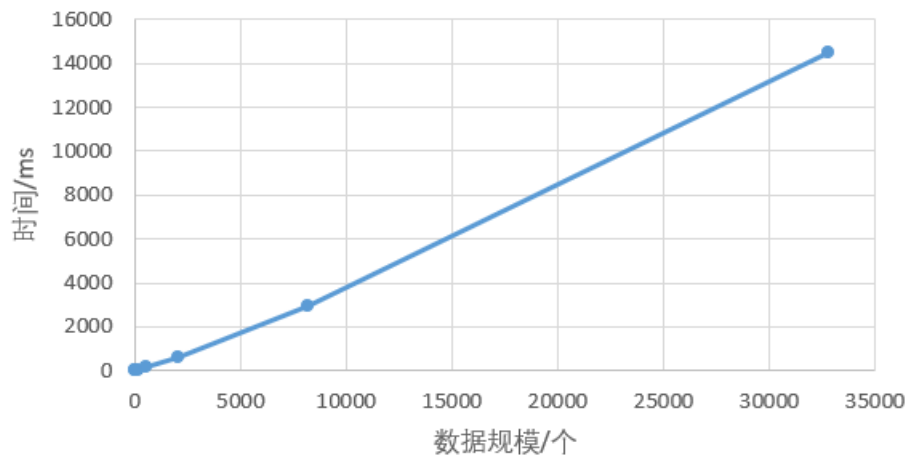
堆排序时间趋势



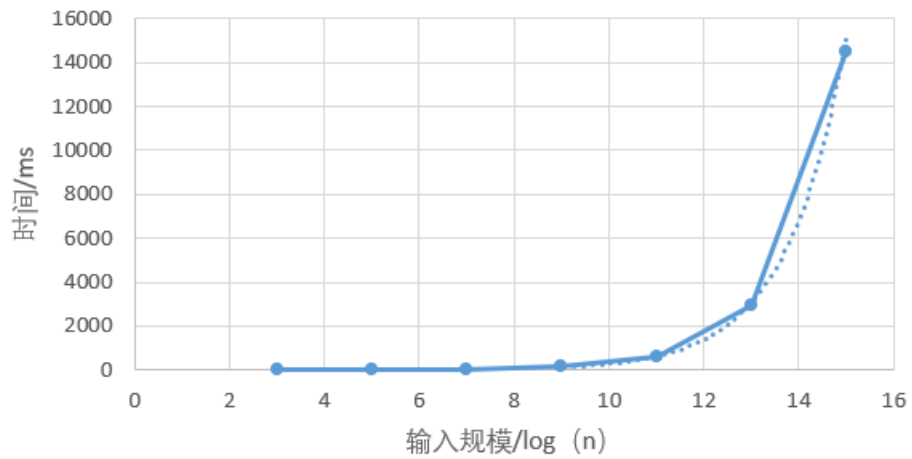
通过两个图像的综合分析，可以看到堆排序排序是的渐进时间大致符合 $\Theta(n \log_2 n)$ 。

### 3>快速排序

快速排序时间趋势

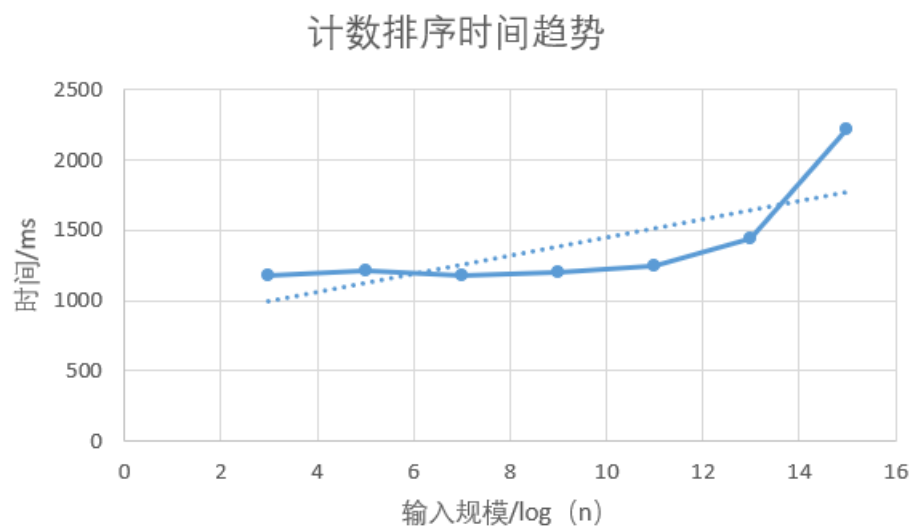
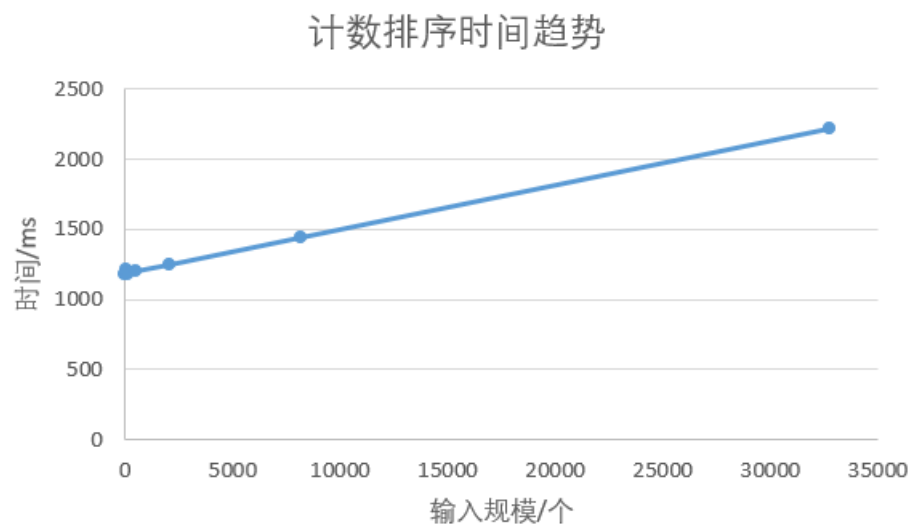


快速排序时间趋势



通过两个图像的综合分析，可以看到快速排序是的渐进时间大致符合 $\Theta(n \log_2 n)$ 。

#### 4>计数排序



通过两个图像的综合分析，可以看到计数排序是的渐进时间不太符合 $\Theta(n + k)$ 。误差较大的原因可能是：

- 1> 数据规模较小，无法对趋势进行准确分析；
- 2> 限制了  $k$  后，计数排序中每次对整个 0-65536 范围的  $C[]$  辅助数组的操作使得在规模很小时，仍有大量时间浪费，使得规模较小的时间偏大。

### 3.结论

可以看到，如果输入规模不大，计数排序的效果不是很理想，但

是归并排序，堆排序，快速排序的效果都很好。如果输入规模变大，计数排序有很大的优势。除此之外可以看到，快速排序的运行时间大致为另外两种的一半，所以：

输入规模较小时，用快速排序，输入规模较大时，尽量用计数排序（需要满足前提条件，即数据范围已知，并满足和  $n$  的关系）。