

稳定的快速排序算法的实现

LUIS LUZERN YUVEN *
数学科学学院，信息与计算科学专业
学号：3190300985

October 20, 2020

1 问题

标准的快速排序算法是不稳定的，但是通过一些方法可以解决这个问题。这里先对数字排序，然后对相同的数字按照原下标排序。

2 实验方案

测试运行环境为虚拟机 Virtual Box 下的 Ubuntu 16.04 。由于测试的是相对时间效率，因此对具体的机器性能不敏感，这里不再列出再多配置细节。

2.1 项目文件说明

程序采用 2011 标准的 C++ 编写，项目名称为 QuickSort，项目文件结构如下：

```
QuickSort
|---generate
|---stat
|---quicksort
```

其中，

generate 能自动产生指定长度提供排序测试的从 1 到 100 的整数，提高出现重复数字的可能性，并采用 C++11 提供 **random** 库确保随机性，其中随机数种子采用对一个具体程序过程的实时统计得到，而时间计算则调用了 C++11 提供的 **chrono** 库，精度到纳秒（ 10^{-9} 秒）。

stat 用于对批量产生时间数据的统计，目前只实现了多次重复测试的平均时间。

quicksort 提供计算一个具体排序运行时间的程序和脚本，其中 **main.cpp** 用于实际时间计算，同样通过 **chrono** 库完成；**bash** 脚本 **batch_test** 共有 4 个用户参数，依次分别代表：

*email : luzernyl@gmail.com

- \$1 测试总组数
- \$2 测试的起始数组长度
- \$3 每组测试的增量
- \$4 每组测试的重复次数

专门用于测试排序的相对时间效率。而脚本 `test4matlab` 功能与参数设置 和 `batch_test` 一致，区别在于其输出为一个 `Matlab` 脚本文件的格式：

3 数值结果

实验结果见图 1。从 t_n 的增长趋势可以看出，随着 $n \log n$ 增大，这一现象与结论：

$$t_n = \theta(n \log n)$$

一致。

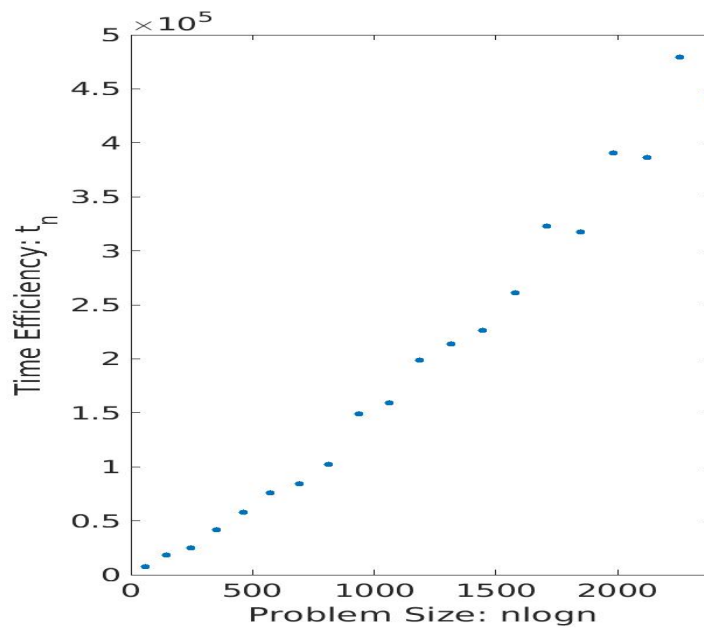


图 1：快速排序对于随机分布整数数组的时间效率，这里每一个点代表一组重复了 100 次后取平均的实验结果。共 20 组，输入数组的规模从长度 20 起，等量递增到 400。每一次重复实验均做了新抽取。

4 结论

实验结果验证了理论结果：插入排序的平均时间效率是 $t_n = \theta(n \log n)$