

插入排序的最坏时间效率测试

LUIS LUZERN YUVEN *

数学科学学院，信息与计算科学专业

学号：3190300985

September 25, 2020

1 问题

通过实际数值算例，统计并验证插入排序在最坏的相对时间效率是

$$t_n = \theta(n^2)$$

其中， n 表示问题的输入规模，在本例中，即指输入的待排序组的数组长度； t_n 表示对一个输入规模为 n 的插入排序过程所耗费的实际时间。对问题和算法的详细描述，详见附件。

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 InsertionSort，项目文件结构如下：

```
InsertionSort
|---C_style
|---generate_data
|---stat
|---timer
|---vector
```

其中，

C_style 为一个标准 C 风格的实现；

*email : luzernyl@gmail.com

`generate_data` 能自动产生指定长度提供排序测试的随机数整数数组，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）。也用了函数 `bigsmall`，把随机数从大到小排序，得到插入排序的最坏情形。

`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。

`timer` 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：

- \$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：

`vector` 为一个 **C++ STL** 风格的实现，其中 `InsertionSort` 被单独写成一个函数 `InsertionSort`，便于精确计算排序时间。本实验实际测试的是这一版本的程序。

2.2 实验步骤

1. 在 `timer` 目录下执行：

```
./test4matlab 20 20 20 100 > result.m
```

产生 20 组数组长度从 20 以等量 20 递增到 400，每组重复测试 100 次，每一次的输入数组均重新抽取。并将结果重定向为 **Matlab** 脚本文件 `result.m`；

2. 在 **Matlab** 或 `octave` 中执行 `result.m` 脚本，观察图像行为；

3. 在 **Matlab** 中运行：

```
plot(re(:, 1), re(:, 2)./re(:, 1).^2, '.', 'MarkerSize', 20);  
xlabel('Problem Size: n');  
ylabel('Time Efficiency : t_n / n^2');  
set(gca, 'FontSize', 40);
```

以观察 $\frac{t_n}{n^2}$ 的极限行为，并将结果调整为便于输出的图像。

4. 分析数据，得出结论并撰写报告。

3 数值结果

实验结果见图 1。从 $\frac{t_n}{n^2}$ 的增长趋势可以看出，随着 n 增大，这一现象与结论：

$$t_n = \theta(n^2)$$

一致。

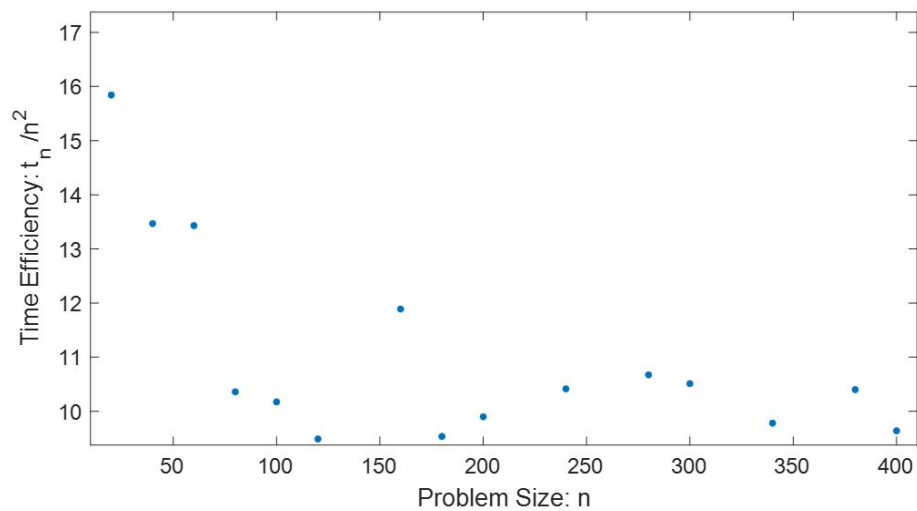


图 1：插入排序对于随机分布整数数组的时间效率，这里每一个点代表一组重复了 100 次后取平均的实验结果。共 20 组，输入数组的规模从长度 20 起，等量递增到 400。每一次重复实验均做了新抽取。

4 结论

实验结果验证了理论结果：插入排序的最坏时间效率是 $t_n = \theta(n^2)$