

华东师范大学数据科学与工程学院上机实践报告

课程名称：算法设计与分析

年级：21 级

上机实践成绩：

指导教师：金澈清

姓名：杨茜雅

上机实践名称：矩阵乘法

学号：

上机实践日期：2022.3.3

10215501435

上机实践编号：No.2

一、目的

1. 熟悉算法设计的基本思想
2. 掌握 Strassen 算法的基本思想，并且能够分析算法性能

二、内容与设计思想

1. 编程实现普通的矩阵乘法；
2. 编程实现 Strassen's algorithm；
3. 在不同数据规模情况下（数据规模 $N=2^3, 2^5, 2^7, 2^9$ ）下，比较两种算法的运行时间各是多少；普通：取平均数后分别是：752, 758.8, 872.6, 1236
Strassen:取平均数后分别是：591.4, 274.4, 500.2, 682.2
4. （附加题）尝试修改 Strassen's algorithm，使之适应矩阵规模 N 不是 2 的幂的情况；

三、使用环境

1. 编译器（比如，在 [Downloads - Mingw-w64](#) 下载一个 GCC / Mingw-w64 版本高一些的编译器。）
2. IDE 或代码编辑器（比如，clion, codeblocks, dev-c++等）

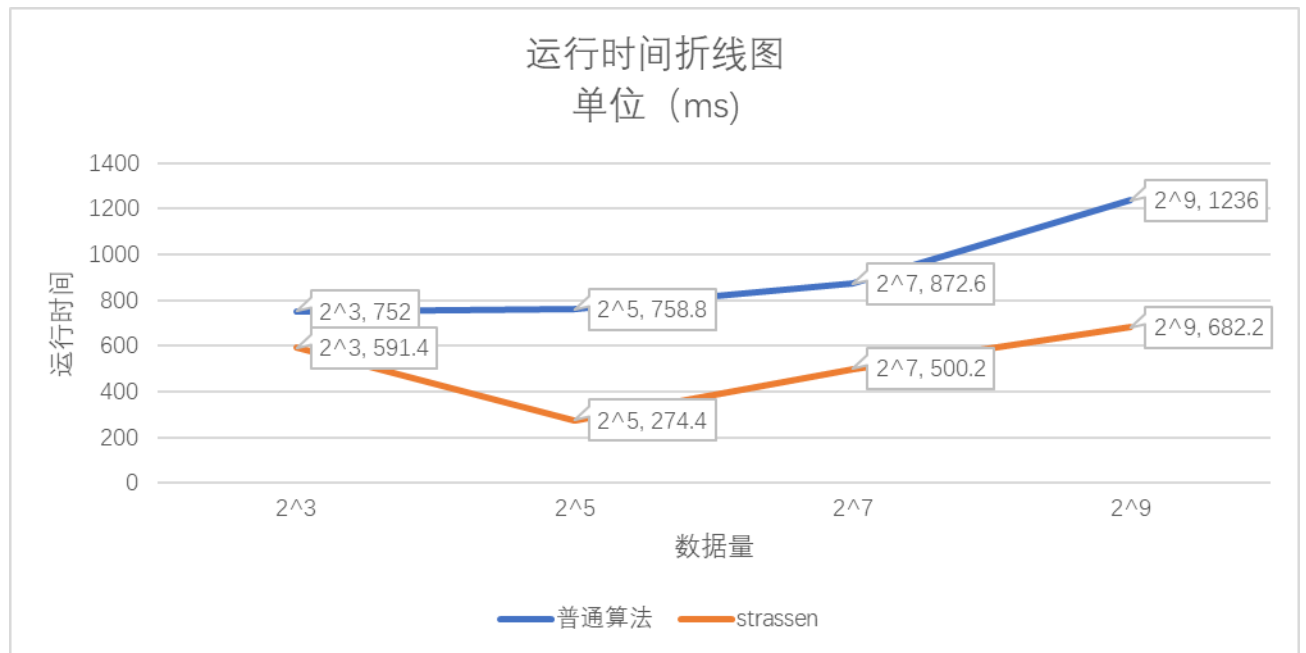
为便于同学操作，我们设置了课程编程题在 [oj](#) | 《算法设计与分析》第二次实验课，同学可以提交算法代码到这个平台去计算和比较时间。

注意 1：每组样例有对应的执行次数，如果需要得到单次时间，要除以重复数。

注意 2：由于平台的缘故，可能会有 30ms 的波动；解决办法是多提交几次（比如 5 次）取平均数。

四、实验过程

1. 写出实验内容中算法的代码，测试，并在 OJ 上提交
2. 收集代码运行时间数据，画出 2 种算法在不同数据量下性能的折线图



3. 总结代码编写时的思路

标准算法：用 `vector` 函数创建多维数组以构造一个一对矩阵相乘的函数，再在主函数中构造出三维数组（需要辨别是否该矩阵是一对）即可

Strassen: 主题是分而治之思想，不断递归对矩阵做分治拆解，简单来说就是把一个大块矩阵不断做 `strassen` 拆解，在 `strassen` 函数内部只做加减法，当拆解到一定大小的矩阵时，将每个小矩阵用普通的矩阵乘法算出来

4. 分析实验内容中算法的复杂度，并说明思路

普通矩阵相乘的复杂度是 $T(n) = 8 T(n/2)$ // 子矩阵的数量 + $\Theta(n^2)$ // 用于子矩阵相加的开销 $T(n) = \Theta(n^3)$. `strassen` 算法 $T(n) = 7 T(n/2)$ // 执行乘法的次数 + $\Theta(n^2)$ // 合并操作的开销（加法）复杂度是 $\Theta(n^{\lg 7})$ 。

其中两者算法都是应用主定理求出

五、总结

对上机实践结果进行分析，问题回答，上机的心得体会及改进意见。

在程序涉及的数据量过于庞大时，选用不合适的算法有可能导致超时。要适时通过判别计算机程序中耗时大的过程并把它替换成耗时小的过程（`strassen` 算法中就是将乘法替换为更快的加法）以减少计算机的负担。