

数据结构与算法实验报告 1——矩阵乘法

软件 52 陈书新 2015013229

csx15@mails.tsinghua.edu.cn

一、实验简介

本实验比较了两个 n 阶方阵相乘的暴力方法和 Strassen 算法的理论复杂度与实际运行时间，随后进一步改进 Strassen 算法，使之在 n 较大时在理论复杂度和实际运行时间两方面均优于普通的矩阵乘法。

二、实验过程

设方阵的阶为 n ，为了实验和编写代码的方便，以下所有的 n 均为 2 的整数次方。

普通的矩阵乘法的时间复杂度为 $\Theta(n^3)$ 。Strassen 算法借鉴了分治的思想，将 n 阶方阵分成 4 个 $n/2$ 阶方阵，进行分块的矩阵乘法运算，如此递归下去。根据主定理，该算法的时间复杂度为 $\Theta(n^{\log_2 7}) \approx \Theta(n^{2.81})$ 。

根据理论的算法复杂度，当 n 较大时，Strassen 算法的运行时间会远远优于暴力方法。

例如当 $n = 512$ 时， $\frac{n^{2.81}}{n^3} = \frac{41025000}{134217728} \approx 0.3$ 。但经过编程并计算运行时间之后，我们得到了下面的表格（单位为秒）。

算法 n	64	128	256	512	1024	2048	4096
Brute-force	0.002	0.014	0.096	0.934	12.398	121.056	1116.911
Strassen	0.051	0.353	2.419	16.527	115.381	805.632	5881.912

我们发现，无论 n 的大小，暴力方法的运行时间总是小于 Strassen 算法，特别是在 n 较大时，Strassen 算法的运行时间已经是暴力方法的 5 倍以上。

我们需要分析产生这种理论时间复杂度与实际运行时间产生偏差的原因，可以联系到两个算法的空间复杂度。暴力方法只需要存储两个待运算的 n 阶方阵和一个结果方阵即可，空间复杂度为 $\Theta(n^2)$ 。Strassen 算法的空间复杂度也可以通过主定理得到。

用 $S(n)$ 表示 Strassen 算法的空间花费，我们有 $S(n) = 7S(\frac{n}{2}) + 2n^2$ ，这个递推式与该算法的时间复杂度递推式类似，可以得出 $S(n) = O(n^{2.81})$ 。由此可以看出，Strassen 算法是一个在用空间换时间的算法。

从递归树可以看出，大量的空间花费集中在递归树接近叶节点的部分，也就是在矩阵很小的时候，我们需要比矩阵较大时使用更多的空间，并且这些空间是不连续的。向内存申请空间花费了大量的时间，而在算法理论时间复杂度计算中，这些申请空间的时间是忽略不计的。

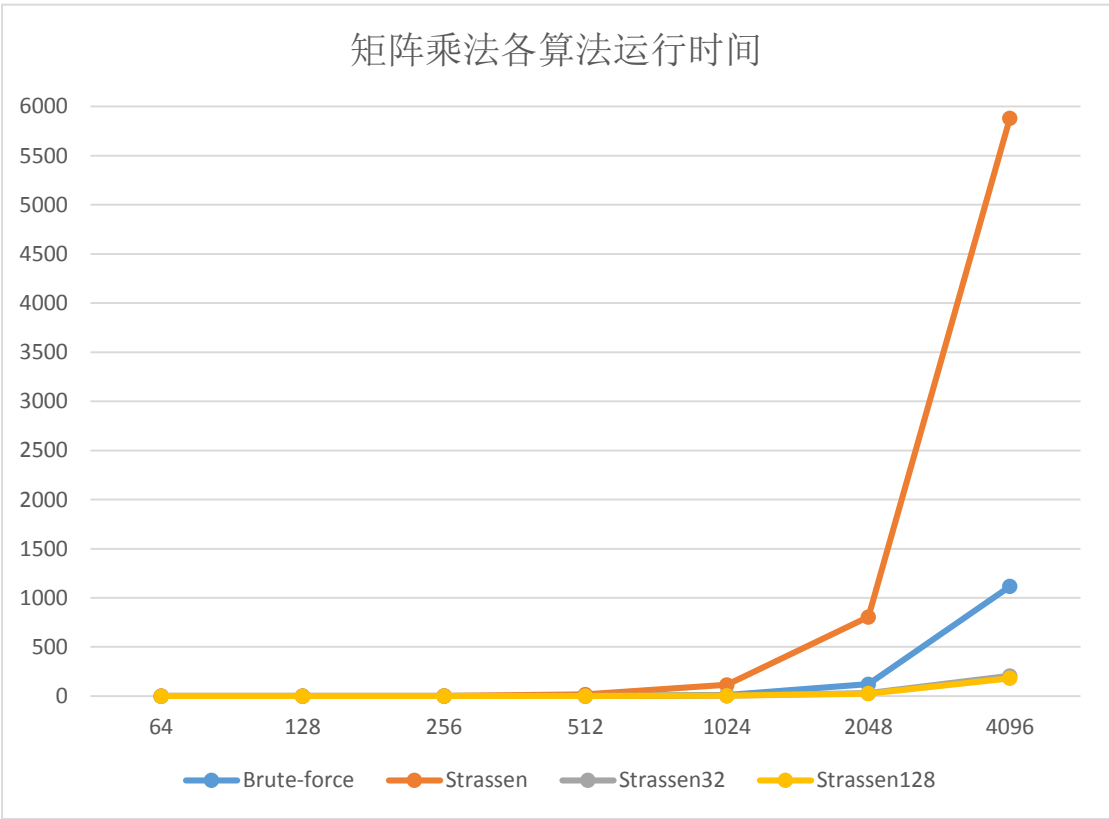
这样，我们可以得到一个优化 Strassen 算法的方法，当 n 较大时，我们仍然用 Strassen 算法递归计算矩阵乘法；当递归到 n 较小时，我们结束递归，使用暴力方法计算结果并返回。

由于 n 是 2 的整数次方，因此我们可以设置一个 2^k 的阈值。当 $n \leq 2^k$ 时使用暴力方法，否则使用 Strassen 算法。

本实验中，我们取了 $k = 1, 5, 7$ 。当 $k = 1$ 时，即为未优化的 Strassen 算法，当 $k = 5, 7$ 时，为阈值为 32, 128 的 Strassen 算法，简称为 Strassen32 和 Strassen128。加上暴力方法，这四个算法的运行时间见下表（单位为秒）。

算法 n	64	128	256	512	1024	2048	4096
Brute-force	0.002	0.014	0.096	0.934	12.398	121.056	1116.911
Strassen	0.051	0.353	2.419	16.527	115.381	805.632	5881.912
Strassen32	0.003	0.016	0.098	0.643	4.245	29.306	206.168
Strassen128	0.003	0.014	0.088	0.561	3.859	25.645	180.966

通过该表格，我们发现加上优化后，Strassen 算法的运行时间大大降低，在 n 较大时运行时间已经低于暴力方法的运行时间。根据上表得到下面的折线图。



通过表格和折线图，我们发现在 $n = 256 \sim 512$ 之间的某一个值为优化后的 Strassen 算法和暴力方法的分界线，此时 Strassen 算法已经适用于 n 较大的情况，实验完成。

三、参考文献

[1] Handawnc. 矩阵相乘常规法和 Strassen 方法算法分析[EB/OL]. <http://blog.csdn.net/handawnc/article/details/7987107>, 2012-09-17.