算法设计与分析实验报告

实验名称：矩阵乘法(分治法)

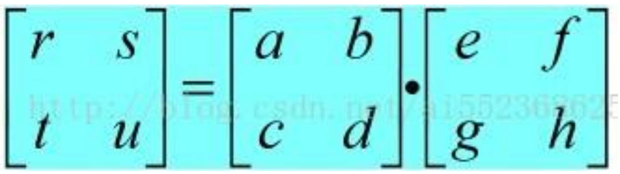
一、问题陈述，相关背景、应用及研究现状的综述分析

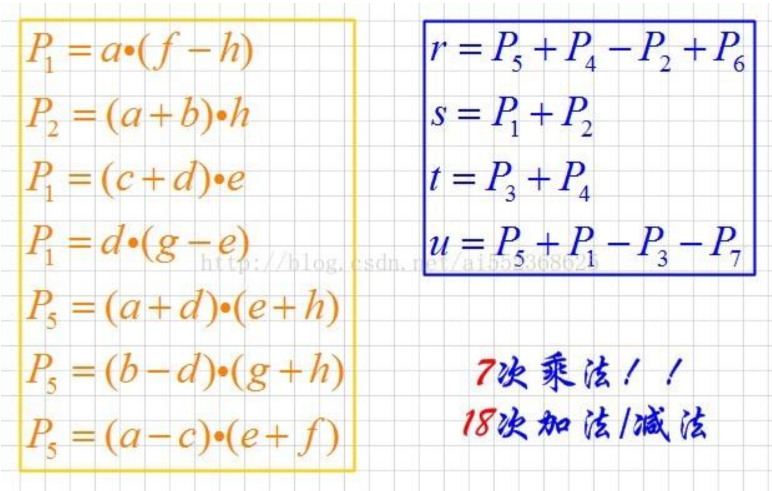
1.问题陈述：

设 A 和 B 是两个 n * n 阶矩阵，求它们的乘积矩阵 C。这里，假设 n 是 2 的幂次方

2.相关背景：

1969 年，Volker Strassen 发表文章提出一种渐进快于平凡算法的矩阵相乘算法，引起巨大轰动。在此之前，很少人敢设想一个算法能渐进快于平凡算法。矩阵乘法的渐近上界自此被改进了。





二、模型拟制、算法设计和正确性证明

1.算法设计：

利用C++类定义一个矩阵类Matrix，然后对于矩阵类重载+，-，*，=运算符来方便计算。然后定义void Strassen(Matrix& A, Matrix& B,Matrix& C)函数，在其中将矩阵A,B,C分块，计算对应的A11,A12,A21,A22,B11,B12,B21,B22,进一步利用递归计算M1到M7，然后计算C11,C12,C21,C22，最后合并成为结果C，递归出口为最简单的2维矩阵相乘，直接调用普通矩阵乘法运算。

三、时间和空间复杂性分析

时间复杂度：

$$T(n) = 7T(n/2) + \Theta(n^2)$$

$$T(n) = \Theta(n^{\lg 7}) \approx \Theta(n^{2.81})$$

空间复杂度：

每次对一个矩阵的分块都会使用四个新的矩阵，新矩阵的维数是原矩阵的一半，但是四个新的矩阵的大小和原矩阵是一样的，所以消耗的新空间和原矩阵的维数有关，设矩阵维数是 2 的 m 次幂，则

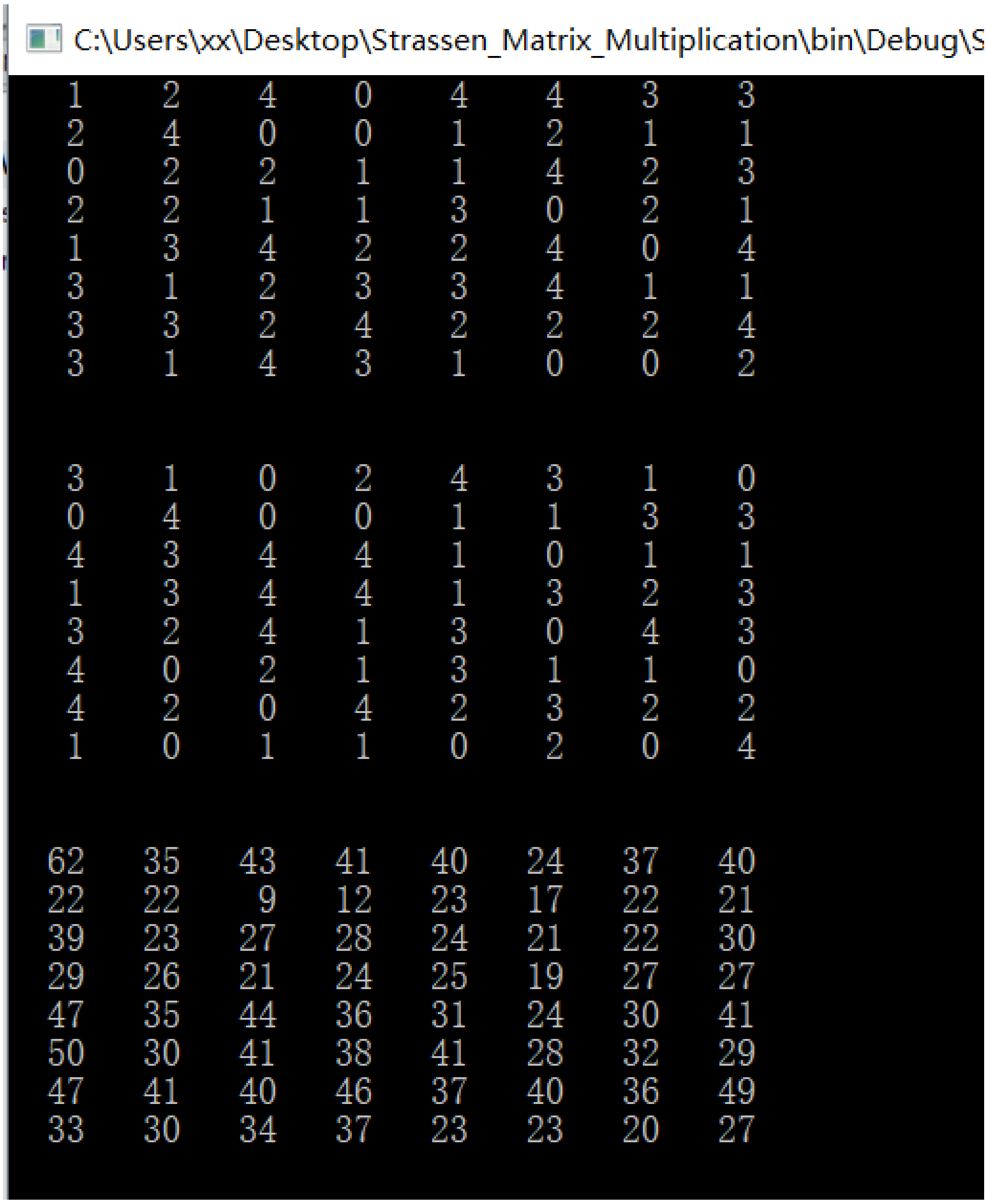
$$S（n）=O(m*3n^2)=O(n^2)$$

四、程序实现和实验测试过程

源程序见 Strassen_Matrix_Multiplication.cpp

测试过程：

测试 8 维矩阵，矩阵内容由随机数定义。



五、总结

对矩阵类重载运算符可以使矩阵运算清晰方便许多。