算法设计与分析实验报告

实验名称:矩阵乘法(分冶法)

- 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近快于平凡算法。矩阵乘法的渐近上界自此被改进了。

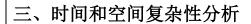
$$\begin{bmatrix} r & s \\ t & u \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

$$P_1 = a \cdot (f - h)$$
 $r = P_5 + P_4 - P_2 + P_6$
 $P_2 = (a + b) \cdot h$ $s = P_1 + P_2$
 $P_1 = (c + d) \cdot e$ $t = P_3 + P_4$
 $P_1 = d \cdot (g - e)$ $u = P_5 + P_1 - P_3 - P_7$
 $P_5 = (a + d) \cdot (e + h)$ $r = (a - c) \cdot (e + f)$ $r = (a - c) \cdot (e + f)$ $r = (a - c) \cdot (e + f)$

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。



时间复杂度:

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

$$\mathsf{T}(\mathsf{n}) = \Theta(\mathsf{n}^{\mathsf{lg7}}) \approx \Theta(\mathsf{n}^{2.81})$$

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。

C:\Users\xx\Deskto	p\Strassen Matrix	Multiplication	\bin\Debua\S
= c. loacia barib carro	p (o crasseri_imacrix	_mancipiicacion	(biii) (bebag (c

- 1			•	1.5	_		- '		 J.
	1 2 0 2 1 3 3	2 4 2 3 1 3	4 0 2 1 4 2 2 4	0 1 1 2 3 4 3	4 1 3 2 3 2 1	4 2 4 0 4 4 2 0	3 1 2 2 0 1 2 0	3 1 3 1 4 1 4 2	
	3 0 4 1 3 4 4	1 4 3 2 0 2 0	0 4 4 4 2 0 1	2 0 4 1 1 4	4 1 1 3 3 2 0	3 1 0 3 0 1 3 2	1 3 1 2 4 1 2 0	0 3 1 3 0 2 4	
	62 22 39 29 47 50 47 33	35 22 23 26 35 30 41 30	43 9 27 21 44 41 40 34	41 12 28 24 36 38 46 37	40 23 24 25 31 41 37 23	24 17 21 19 24 28 40 23	37 22 22 27 30 32 36 20	40 21 30 27 41 29 49 27	

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。