Program Assignment 2 **Longxiang Li**
CS124 Teammates: No

Introduction

Strassen's matrix multiplication algorithms works faster for n by n matrix than the conventional naive algorithm whose time complexity is $O(n^3)$, in theory. However, for some smaller n, the conventional algorithm runs faster because the time consumed in memory allocation in Strassen's algorithm. But, for this recursive algorithm, we don't need to go to the basic element of this matrix (a 1×1). We may modified the classic algorithm by converting from recursion to conventional algorithm when the dimension is small enough. We call this cross-over point. In this experiment, we try to find the optimal cross-over point for different n from two direction. First, we calculate the exact running time for Strassen's algorithm, modified Strassen's algorithm and conventional algorithm and estimate the cross-over point numerically. These calculation were based on the assumption that the cost of any single arithmetic operation is 1 and all others are free. Second, we implemented these algorithms and get the cross-over point in practice.

Numerical Calculation

To simplify the calculation we assume in the following analysis in this section, the dimension n is power of 2 and the cross-over value is also power of 2. In classic Strassen's algorithm, for a given matrix of given n, we require 7 multiplication of matrices of simension $\frac{n}{2}$. In addition, there are 18 summation and subtraction of size $\frac{n}{2}$. Thus, we can write a recursive equation of the running time:

$$\begin{aligned} f_1(n) &= 7f(\frac{n}{2}) + 18(\frac{n}{2})^2 \\ &= 7^2 f(\frac{n}{4}) + 7 \times 18(\frac{n}{4})^2 + 18(\frac{n}{2})^2 \\ &= 7^k f(\frac{n}{2^k}) + 18n^2\{7^{k-1}(\frac{n}{2^k})^2 + \ldots + 7^0(\frac{n}{2})^2\} \end{aligned}$$

when $k = log_2 n$, we arrive at the basic case. Since $f(1) = 1$. We can conclude that the exact running time of standard Strassen's algorithm is:

$$f_s(n) = 7^{log_2 n} + 6n^2\{(\frac{7}{4})^{log_2 n} - 1\}$$

For conventional algorithm, we have a total of n^3 multiplication and $n^2(n-1)$ additions. So the close form running time of conventional algorithm is:

$$f_c(n) = n^2(2n-1)$$

Let's assume we set cross-over point at n_0. That means above n_0, we used Strassen's algorithm and below n_0, we used conventional algorithm. So the running time of this hybrid algorithm is:

$$f_{n_0} = 7^{log_2 \frac{n}{n_0}}(2n_0^3 - n_0^2) + 6n^2((\frac{7}{4})^{log_2 \frac{n}{n_0}} - 1)$$

Alternative Solutions

Optimum Solution

Construction/Implementation

Analysis & Testing

Final Evaluation

Attachments

	n	Strassen's	2*2	4*4	8*8	16*16	Conventional
4	16.00	15271.00	10812.00	14928.00	29376.00	65280.00	7936.00
5	32.00	111505.00	80292.00	109104.00	210240.00	461568.00	64512.00
6	64.00	798967.00	580476.00	782160.00	1490112.00	3249408.00	520192.00
7	128.00	5666497.00	4137060.00	5548848.00	10504512.00	22819584.00	4177920.00
8	256.00	39960391.00	29254332.00	39136848.00	73826496.00	160032000.00	33488896.00
9	512.00	280902385.00	205959972.00	275137584.00	517965120.00	1121403648.00	268173312.00
10	1024.00	1971035287.00	1446438396.00	1930681680.00	3630474432.00	7854544128.00	2146435072.00
11	2048.00	13816121377.00	10143943140.00	13533646128.00	25432195392.00	55000683264.00	17175674880.00
12	4096.00	96788347111.00	71083099452.00	94811020368.00	178100865216.00	385080280320.00	137422176256.00