Matrix Multiplication Algorithm

Nhóm thực hiện

1611657 – Nguyễn Minh Khôi

1611288 – Nguyễn Khắc Quang Huy

1612372 – Võ Trung Nhân

Môn Cấu trúc Dữ liệu và Giải Thuật

GV: Dr. Trần Tuấn Anh

# TỔNG QUAN VỀ THUẬT TOÁN NHÂN MA TRẬN

Trong toán học, chúng ta đã biết về phương pháp tìm ma trận là kết quả của phép nhân 2 ma trận khác. Nhưng trong Khoa học Máy tính, các chuyên gia còn tìm ra nhiều cách khác để thực hiện nhu cầu đó với kích thước ma trận khổng lồ trên máy tính.

# TESTING AND TIMING

* Để thực hiện việc test, nhóm đưa ra 12 bộ dữ liệu, có kích thước 1024x1024, gồm 3 nhóm dữ liệu (int, float và double), mỗi nhóm có 4 bộ dữ liệu khác nhau.
* Công cụ đếm thời gian: sử dụng package <chrono> của C++11 để tăng khả năng chính xác lên miliseconds so với package <ctime>.

# KẾT QUẢ

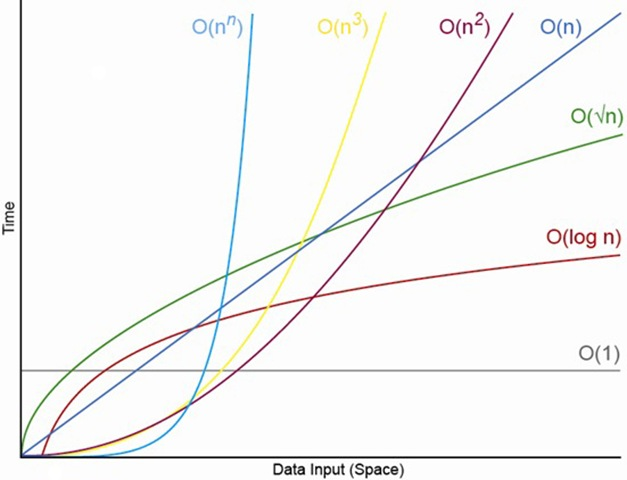
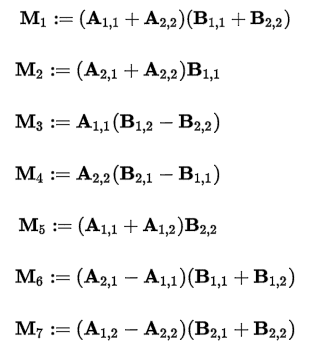


Figure 1: Time Consumption on each Big-O notation

## Thuật toán Matrix Iterative

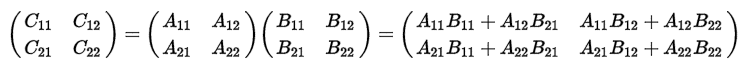
Đây là thuật toán cơ bản được vận dụng trong toán học để tính phép nhân ma trận. Thuật toán này tốn thời gian chạy trên lý thuyết.

## Thuật toán Strassen



Thuật toán này chí tốn . Với kích thước ma trận cơ sở là 2x2, thông thường mất 8 bước để tính kết quả, thì với Strassen, chúng ta chỉ cần 7 bước như hình trái; sau đó các giá trị thành phần của ma trận kết quả sẽ được tính như hình phải.

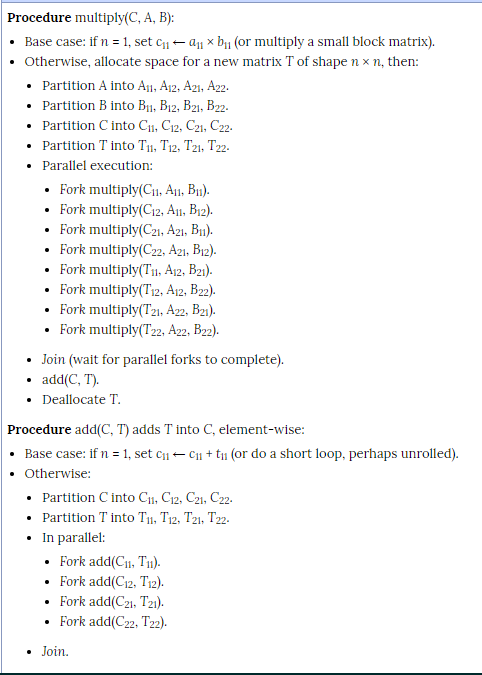
## Thuật toán “Chia để trị”

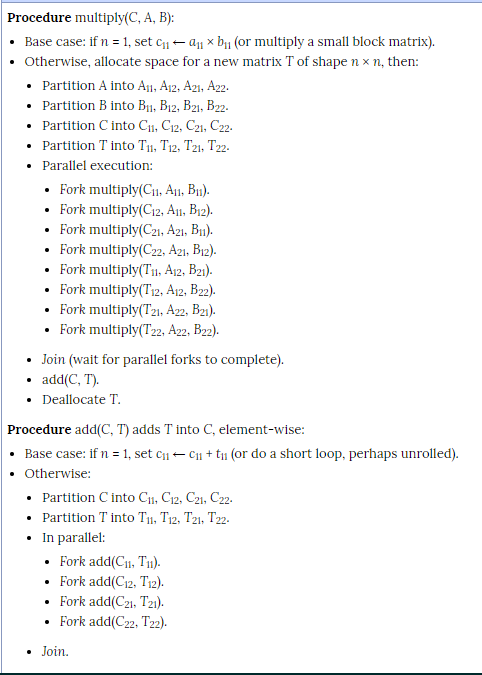


Thuật toán này dùng ma trận 2x2 làm cơ sở (nên các ma trận nhập phải có kích thước là lũy thừa của 2). Các ma trận nhập vô sẽ được chia làm 4 và đệ quy liên tục cho đến khi các ma trận con đạt kích thước cơ sở và được tính toán như hình trên.

## Thuật toán song song

Thuật toán này có nhiều điểm tương đồng với thuật toán Strassen, nhưng sử dụng tính song song, so với Strassen là tuần tự.

Thuật toán này cũng lấy ma trận 2x2 làm cơ sở. Ma trận nhập phải có kích thước là lũy thừa của 2, sau khi đưa vào thuật toán sẽ được phân rã liên tục thành các ma trận con có kích thước giảm đi một nửa so với ma trận ban đầu và tiến hành thuật toán như hình dưới. Thuật toán này cần hỗ trợ lập trình song song.



## Thuật toán Cannon

Thuật toán này cốt lõi tương đồng với Matrix Iterative, nhưng đồng thời trong quá trình nhân, có thêm phần dịch chuyển dữ liệu các phần tử đầu vào. Thuật toán này cần hỗ trợ lập trình song song.

# BÀN LUẬN

Bảng kết quả thời gian (giây) test thuật toán, bộ dữ liệu kích thước 1024x1024   
1-4 (kiểu int), 5-8 (kiểu float) và 9-12 (kiểu double).

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Naive | 21.74 | 22.27 | 21.78 | 21.66 | 21.31 | 21.32 | 22.86 | 26.63 | 28.95 | 25.87 | 25.39 | 25.47 |
| D&C | 79.84 | 80.69 | 79.92 | 79.83 | 82.11 | 80.96 | 80.86 | 81.09 | 82.15 | 81.52 | 84.66 | 81.66 |
| Strassen | 145.35 | 146.59 | 144.38 | 147.26 | 150.95 | 150.10 | 150.27 | 159.80 | 144.85 | 144.96 | 144.93 | 144.89 |

\*D&C: Chia để trị

## Thuật toán Strassen

### GIẢI THÍCH LÝ DO

Thuật toán Strassen trên lý thuyết là , nhưng khi kiểm tra thực tế, thời gian thực hiện nhiều hơn so với thuật toán toán học thông thường, vốn trên lý thuyết là ; bởi Strassen gọi hàm đệ quy nhiều lần, tốn bộ nhớ và thời gian sao chép các giá trị, giảm hiệu suất của thuật toán. Thuật toán này đòi hỏi bộ nhớ lớn để xử lý. Khả năng với bộ dữ liệu lớn hơn và bộ nhớ lớn hơn, thuật toán Strassen sẽ cải thiện đáng kể được tốc độ.

## Thuật toán “Chia để trị”

### GIẢI THÍCH LÝ DO

Thuật toán “Chia để trị” chậm hơn so với thuật toán bình thường (Naive) và nhanh hơn Strassen. Lý do ở đây là do quy hồi khá nhiều trong quá trình tính toán (ít hơn Strassen) nên đã giảm tốc độ của thuật toán. Tuy nhiên đây vẫn là thuật toán .

Table of Contents

[TỔNG QUAN VỀ THUẬT TOÁN NHÂN MA TRẬN 1](#_Toc492234450)

[TESTING AND TIMING 1](#_Toc492234451)

[KẾT QUẢ 1](#_Toc492234452)

[Thuật toán Matrix Iterative 1](#_Toc492234453)

[Thuật toán Strassen 1](#_Toc492234454)

[Thuật toán “Chia để trị” 1](#_Toc492234455)

[Thuật toán song song 1](#_Toc492234456)

[Thuật toán Cannon 1](#_Toc492234457)

[BÀN LUẬN 1](#_Toc492234458)

[Thuật toán Strassen 1](#_Toc492234459)

[GIẢI THÍCH LÝ DO 1](#_Toc492234460)

[Thuật toán “Chia để trị” 1](#_Toc492234461)

[GIẢI THÍCH LÝ DO 1](#_Toc492234462)