ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH

**TÍNH TOÁN SONG SONG (CO3067)**

---

## Project Introduction

# Đề 2: Nhân ma trận dùng giải thuật Strassen

---

GVHD: Thoại Nam
SV: Nguyễn Hữu Thiên Phú - 1810433
Trương Công Thành - 1810766
Nguyễn Minh Anh - 1811431
Đoàn Ngọc Thịnh - 1810542

Tp. Hồ Chí Minh, Tháng 3/2021

# Mục lục

# 1 Nhân ma trận

## 1.1 Giới thiệu về bài toán nhân ma trận

Nhân ma trận là một bài toán kinh điển, là một tác vụ quan trọng trong nhiều tính toán, giải thuật và nhóm giải thuật như xử lý ảnh, xấp xỉ nghiệm cho hệ phương trình tuyến tính, discrete Fourier transform, ...[2] Đặc biệt là ứng dụng của nhân ma trận trong phương pháp tính (numerical analysis), giúp xấp xỉ nghiệm cho những phương trình, bài toán mà các giải thuật thông thường chỉ có thể giải trên lý thuyết, trên giấy.

Giải thuật nhân ma trận đơn giản nhất dựa trên chính định nghĩa của nó:

```
input: matrix A(nxm), B(mxp)
output: matrix C = AxB

C = 0(nxp); // initialize C to zero

for (int i = 0; i < n; ++i)
    for (int j = 0; j < p; ++j)
        for (int k = 0; k < m; ++k)
            C[i][j] = C[i][j] + A[i][k] * B[k][j];

return C;
```

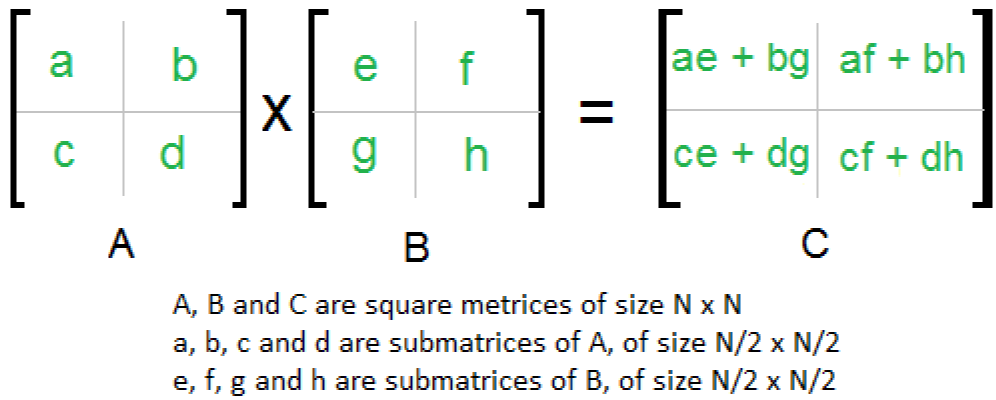
Dễ thấy giải thuật có độ phức tạp về mặt thời gian là $(nmp)$, hay, thuận tiện hơn, $O(n^3)$.

Thế nhưng trong những giải thuật quan trọng - tạo nền tảng cho thời đại số này, nhân ma trận trong $O(n^3)$ lại chính là tác vụ gây ra điểm nghẽn (bottleneck), khiến nhiều bài toán còn chưa thể giải. Điều đó đã trở thành động lực cho nhiều nghiên cứu, công nghệ ra đời nhằm mục đích đưa ra những giải thuật, phương pháp nhân 2 ma trận trong thời gian nhanh hơn. Trong đó, có 2 hướng chủ đạo: nghiên cứu và xây dựng giải thuật nhân ma trận mới, và song song hóa các giải thuật này.

## 1.2 Giải thuật Strassen

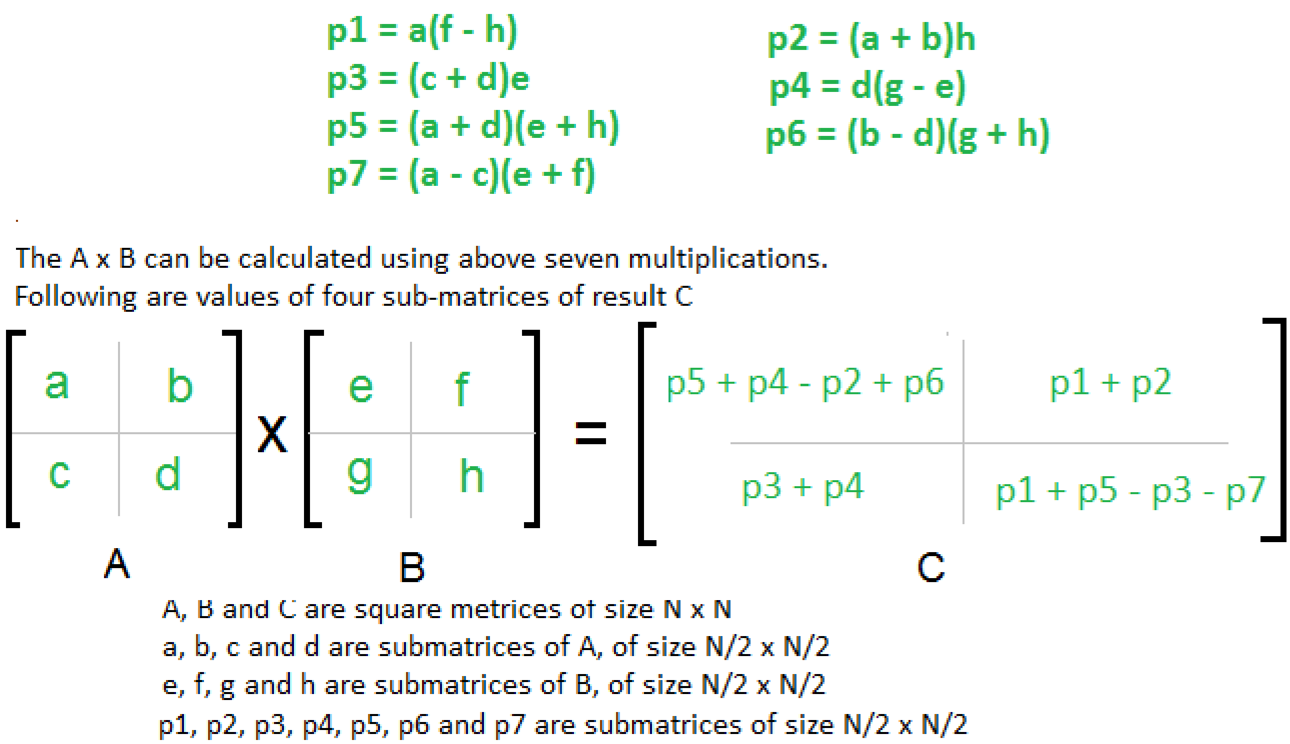
Mở đầu cho hướng thứ nhất, chắc chắn phải kể đến công của Volker Strassen, người đã chứng minh nhân ma trận trong $O(n^3)$ là không tối ưu, đồng thời cho ra đời giải thuật nhân ma trận nhanh đầu tiên, chạy trong $O(n^{log_2(7)}) \approx O(n^{2.8074})$, sử dụng chiến thuật chia để trị (divide and conquer).

Giải thuật Strassen dựa trên quan sát rằng tích AxB, với A, B là 2 ma trận vuông $2^n \times 2^n$, có thể phân tích thành:




Đây là một giải thuật chia để trị ngây thơ (naive) mà Strassen đã dùng để xây dựng giải thuật của mình. Phân tích giải thuật này, ta thấy có 8 phép nhân và 4 phép cộng được thực hiện cho ma trận $2^{n-1} \times 2^{n-1}$, hay $T(N) = 8T(N/2) + 4N^2 = O(N^3)$, không cho ta một cải thiện đáng kể (thậm chí còn chậm hơn do thực hiện đệ quy và sử dụng thêm bộ nhớ).

Nhưng từ giải thuật chia để trị này, Strassen nhận thấy rằng, ta không cần đến 8 phép nhân cho mỗi bước đệ quy, mà chỉ cần 7:




$T(N) = 7T(N/2) + 18N^2 = O(N^{2.8074})$

Đây chính là phát hiện đánh dấu mốc cho ra đời sau đó rất nhiều giải thuật khác cố gắng cải thiện hằng số mũ trong độ phức tạp tính toán.

## 1.3 Tính toán song song

Giải thuật của Strassen là một bước ngoặc lớn, song nó có một overhead không nhỏ: 4 phép cộng thành 18 phép cộng/trừ ma trận. Đây là vấn đề chung của nhóm giải thuật nhân ma trận nhanh (giải thuật Strassen và các cái tiến khác): overhead đáng kể, khó đưa vào áp dụng thực tiễn. Như giải thuật Strassen chỉ cho những cải thiện đáng kể với n $>$ 1,000, nhưng những cải thiện đó cũng chẳng đủ khiến bài toán nhân ma trận với n $>$ 1,000 khả thi hơn[3]. Để giải quyết vấn đề này, người ta không tìm một giải thuật mới, mà rẽ sang một hướng thứ hai: song song hóa những giải thuật tuần tự đã có.

Với sự ra đời của bộ xử lý đa nhân, GPU và các hệ thống nhiều bộ xử lý, bài toán nhân ma trận với n $>$ 1,000, thậm chí 10,000 hay 100,000, trở nên khả thi hơn bao giờ hết. Điều này giúp đưa điểm vượt (crossover point) của giải thuật Strassen so với giải thuật ngây thơ vào vùng khả thi.

### 1.4 Tổng kết

Chính vì lẽ đó, đề tài đã được lựa chọn, đánh giá xem vai trò, hiệu quả và điểm yếu của giải thuật Strassen song song (và một số biến thể cải tiến nhỏ) so với phiên bản tuần tự của chính nó, phiên bản song song cho ngây thơ và một số biến thể (variation) khác. Việc đánh giá sẽ tập trung vào hiệu năng tính toán trên một môi trường duy nhất, do đó ít tập trung vào hiệu năng nhờ caching, các công nghệ khác nhau, hay các thông số khác như bộ nhớ sử dụng, sai số (numerical stability), ... dù chúng có những quan hệ nhất định với hiệu năng tính toán.

## 2 Nguồn dữ liệu

- Nhân ma trận vuông kích thước lũy thừa 2 có kích thước lớn: Giải thuật Strassen được chứng minh sẽ vượt trội so với giải thuật ngây thơ khi n đủ lớn, vậy trước tiên, ta cần tìm xem đâu là crossover point của Strassen song song so với Strassen tuần tự và ngây thơ song song (và ngây thơ tuần tự).
- Nhân ma trận vuông kích thước lũy thừa 2 có kích thước nhỏ: Giải thuật Strassen chậm khi n đủ nhỏ, nhưng chậm hơn bao nhiêu ta sẽ phải đo đạc và đánh giá.
- Nhân ma trận kích thước bất kỳ: Việc giải thuật Strassen dựa trên tính chất kích thước ma trận là lũy thừa 2 đồng nghĩa với việc giải thuật (biến thể cho ma trận bất kỳ) sẽ kém hiệu quả hơn, và cần đo đạc và đánh giá độc lập.

## 3 Môi trường tính toán

Việc tính toán sẽ được thực hiện trên node của server của HPC Lab, bộ xử lý Intel® Xeon® E5-2680 v3, 2.5GHz, song song hóa bằng OpenMP C++. Do server là tài nguyên được chia sẻ cho nhiều user và process, việc đo đạc sẽ được thực hiện nhiều lần và lấy trung bình để giảm thiểu sai số.

## Tài liệu

[1] GeeksforGeeks. 2021. Divide and Conquer | Set 5 (Strassen's Matrix Multiplication) - GeeksforGeeks. Available at:

urlhttps://www.geeksforgeeks.org/strassens-matrix-multiplication/ [Accessed 29 March 2021].

[2] Hardesty, L., 2013. Explained: Matrices. MIT News | Massachusetts Institute of Technology. Available at: https://news.mit.edu/2013/explained-matrices-1206 [Accessed 29 March 2021].

[3] D'Alberto, Paolo; Nicolau, Alexandru (2005). Using Recursion to Boost ATLAS's Performance. Available at: https://www.ics.uci.edu/~paolo/Reference/paoloA.ishp-vi.pdf.