BÁO CÁO THỰC HÀNH MÔN PHÂN TÍCH THUẬT TOÁN

Lab 5 - Tuần 9

Đinh Anh Huy - 18110103

Bài toán 1. Viết chương trình nhân 2 ma trận A và B có kích thước là $n \times n$ với 2 phương pháp như sau:

1. Nhân 2 ma trận như các em đã học trong đại số tuyến tính với độ phức tạp là $O(n^3)$.
2. Dùng kỹ thuật **Chia để trị** để nhân 2 ma trận với độ phức tạp là $O(n^{\log 7})$.

Trong đó

- **Input**: A, B được tạo ngẫu nhiên trong đoạn $[1, 1000]$.
- **Output**: $C = A \cdot B$.
- Chứng minh độ phức tạp của 2 phương pháp do em cài đặt. Vẽ đồ thị so sánh 2 phương pháp với $n = 2^k, k = 10, 11, ..., 32$.

Lời giải

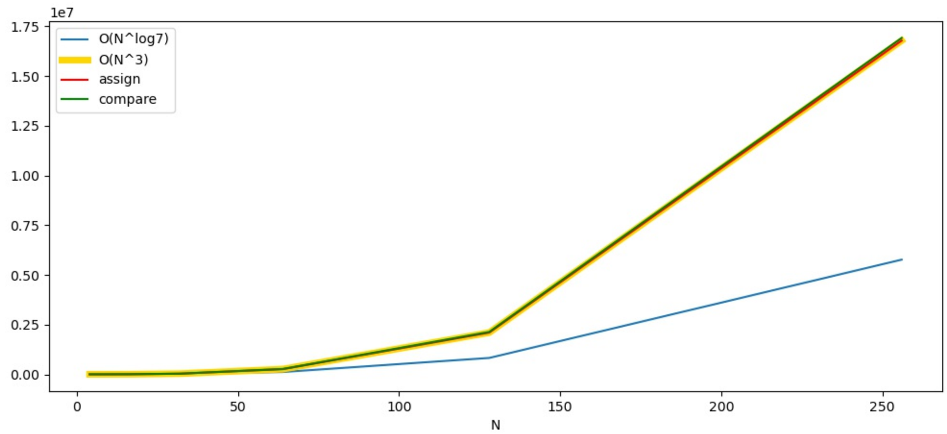
1. Nhân 2 ma trận như các em đã học trong đại số tuyến tính với độ phức tạp là $O(n^3)$.

   Ý tưởng: $A = (A_{ij}), B = (B_{ij})$ thì $C = A \cdot B = \sum_{k=1}^{n} A_{ik} \cdot B_{kj}$, trong đó $i, j = \overline{1, n}$.

Algorithm 1: Nhân 2 ma trận A, B

```
Function Multiply-Matrices(A, B):
    C ← empty array;
    N ← len(A);
    for i = 0 → N do
        for j = 0 → N do
            C[i][j] ← 0;
            for k = 0 → N do
                C[i][j] ← C[i][j] + A[i][k] + B[k][j];
            end
        end
    end
    return C;
```

Ta dễ dàng thấy được thuật toán trên có độ phức tạp là $O(n^3)$ khi có số lần lặp là n^3. Ta có thể kiểm chứng thông qua sự tương đồng giữa các đường biểu thị số phép gán và so sánh của thuật toán trên với đường $O(n^3)$ như sau



2. Dùng kỹ thuật **Chia để trị** để nhân 2 ma trận với độ phức tạp là $O(n^{\log 7})$.

   Ý tưởng: Sử dụng thuật toán nhân 2 ma trận của *Strassen*. Giả sử ta có $n = 2^q$ và 2 ma trận $A \in \mathbb{R}^{n \times n}$ và $B \in \mathbb{R}^{n \times n}$. Nếu $n_{min} = 2^d$ với $d \leq q$ thì thuật toán tính giá trị $C = AB$ bằng cách áp dụng thuật toán đệ quy Strassen.

Algorithm 2: Nhân 2 ma trận A, B bằng thuật toán Strassen

```
Function Strass(A, B, n, n_min):
    if n ≤ n_min then
        C = AB    (conventionally computed)
    end
    else
        m ← n/2;
        u ← 1 : m;
        v ← (m + 1) : n;
        P_1 ← Strass(A(u,u) + A(v,v), B(u,u) + B(v,v), m, n_min);
        P_2 ← Strass(A(v,u) + A(v,v), B(u,u), m, n_min);
        P_3 ← Strass(A(u,u), B(u,v) − B(v,v), m, n_min);
        P_4 ← Strass(A(v,v), B(v,u) − B(u,u), m, n_min);
        P_5 ← Strass(A(u,u) + A(u,v), B(v,v), m, n_min);
        P_6 ← Strass(A(v,u) − A(u,u), B(u,u) + B(u,v), m, n_min);
        P_7 ← Strass(A(u,v) − A(v,v), B(v,u) + B(v,v), m, n_min);
        C(u,u) = P_1 + P_4 − P_5 + P_7;
        C(u,v) = P_3 + P_5;
        C(v,u) = P_2 + P_4;
        C(v,v) = P_1 + P_3 − P_2 + P_6;
    end
    return C;
```

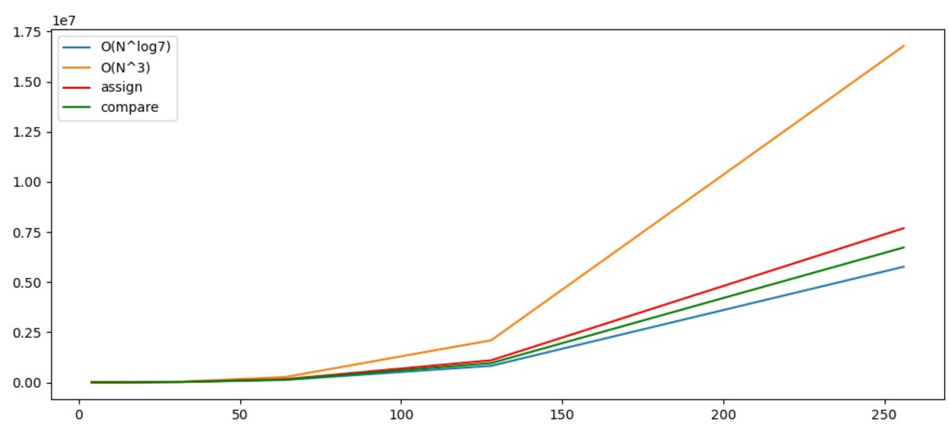
Từ thuật toán trên ta có công thức truy hồi

$$T(n) = \begin{cases} C_1 & \text{nếu } n = 1 \\ 7T(\frac{n}{2}) + C_2 & \text{nếu } n \geq 1 \end{cases}$$

Khi đó, với $k = \log_2 n$, ta có

$$
\begin{aligned}
T(n) = T(\overline{a_k a_{k-1} \ldots a_0}_2) &= 7T(\overline{a_k a_{k-1} \ldots a_1}_0) + C_2 \\
&= 7^2 T(\overline{a_k a_{k-1} \ldots a_2}_2) + 7C_2 + C_2 \\
&\vdots \\
&= 7^k T(\overline{a_k}_2) + C_2 \sum_{i=0}^{k-1} 7^i \\
&= 7^k T(1) + C_2 \sum_{i=0}^{k-1} 7^i \\
&= 7^k C_1 + C_2 \sum_{i=0}^{k-1} 7^i \\
&\leq 7^k C_1 + C_2 \sum_{i=0}^{k-1} 7^k \\
&= 7^k C_1 + 7^k C_2 k \\
&= 7^k (C_1 + kC_2) \\
&= 7^{\log_2 n} (C_1 + kC_2) \\
&= n^{\log_2 7} (C_1 + kC_2) = O(n^{\log_2 7})
\end{aligned}
$$

Do đó, độ phức tạp của thuật toán trên là $O(n^{\log_2 7})$. Ta cũng có thể kiểm chứng thông qua sự tương đồng về hình dáng của các đường biểu thị số phép gán và số phép so sánh của thuật toán trên với đường $O(n^{\log_2 7})$ như sau



Ta có kết quả thực thi của 2 thuật toán như sau

```
               Exercise 1
****************************************
>> Matrix A:
 [[942 555 871 535]
 [543 167 492 779]
 [527 988 238 948]
 [300 913 122 723]]
>> Matrix B:
 [[109  47 651 539]
 [858 523 791 399]
 [ 71 658 749 615]
 [753 781 599 547]]

@ Classic Method
>> Result of multiplication between A and B:
 [[1043564 1325492 2025091 1557493]
 [ 823992 1044997 1320719 1088003]
 [1635889 1438485 1870699 1343191]
 [1369135 1136538 1441938  996498]]

@ Strassen Method
>> Result of multiplication between A and B:
 [[1043564 1325492 2025091 1557493]
 [ 823992 1044997 1320719 1088003]
 [1635889 1438485 1870699 1343191]
 [1369135 1136538 1441938  996498]]

@ Compare with result of built-in function:
 [[1043564 1325492 2025091 1557493]
 [ 823992 1044997 1320719 1088003]
 [1635889 1438485 1870699 1343191]
```

Bài toán 2 (Bonus). Có thể nhân 2 ma trận $n \times n$ với độ phức tạp là $O(n^2)$ không? Nếu có thì chứng minh và cài đặt thuật toán. Ngược lại, giải thích nguyên nhân vì sao không thể nhân 2 ma trận có độ phức tạp như vậy.

Lời giải

Không thể nhân 2 ma trận $n \times n$ với độ phức tạp là $O(n^2)$ vì mỗi ma trận có kích thước $n \times n$ thì sẽ tốn ít nhất $O(n^2)$ để xét từng phần tử của các ma trận, và mỗi phần tử phải được xét ít nhất 1 lần để cho ra kết quả mong muốn, do đó sẽ không thể tồn tại thuật toán nhân 2 ma trận nào có độ phức tạp nhỏ hơn $O(n^2)$ được. Tuy nhiên, tính đến thời điểm hiện tại, thuật toán nhân 2 ma trận có độ phức tạp thấp nhất được phát hiện là thuật toán Coppersmith-Winograd với độ phức tạp là $O(n^{2.3737})$, cho nên thuật toán nhân 2 ma trận có độ phức tạp $O(n^2)$ là không có. Đối với trường hợp 2 ma trận có dạng đặc biệt như ma trận đường chéo thì vẫn tồn tại thuật toán như trên.