Phân Tích Thuật Toán

Bùi Thị Thanh Phương

Ngày 2 tháng 5 năm 2023

Bài 1
Viết chương trình nhân 2 ma trận A và B có kích thước là nn 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 $0(n^{log 7})$
trong đó:

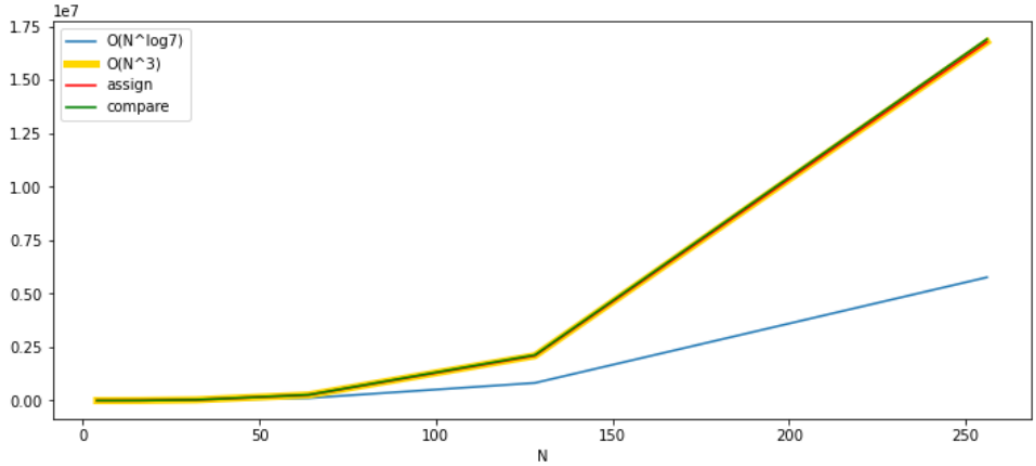
- Input:A, B được tạo ngẫu nhiên trong đoạn $[1, 1000]$
- Output: C = A · B.
- Chứng minh độ phức tạp của 2 phương pháp do em cài đặt. Vẽ đồ thị so sánh 2 phương pháp với $n = 2^k, k = 10, 11, ..., 32$.

Trả lời
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 $A = A_{ij}, B = B_{ij}, thC = A.B = \sum_{k=1}^{n} A_{ik}.B_{kj}$ trong đó $i, j = 1, \ldots, n$

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 for
        end for
    end for
    return C;
end function
```

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^3 như sau
2. Dùng kỹ thuật Chia để trị để nhân 2 ma trận với độ phức tạp là $0(n^{log 7})$



Ý tưởng: Sử dụng thuật toán nhân 2 ma trận của Strassen. Giả sử ta có $n = 2^q$ và 2 ma trận $A \in R^{nxn}$ và $B \in R^{nxn}$. Nếu $n_{min} = 2^d$ với $d \leq q$ 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, nmin)
    if n ≤ nmin then
        C ← A × B (conventionally computed)
    else
        m ← n/2
        u ← 1 : m
        v ← (m + 1) : n
        P1 ← Strass(A(u,u) + A(v,v), B(u,u) + B(v,v), m, nmin)
        P2 ← Strass(A(v,u) + A(v,v), B(u,u), m, nmin)
        P3 ← Strass(A(u,u), B(u,v) − B(v,v), m, nmin)
        P4 ← Strass(A(v,v), B(v,u) − B(u,u), m, nmin)
        P5 ← Strass(A(u,u) + A(u,v), B(v,v), m, nmin)
        P6 ← Strass(A(v,u) − A(u,u), B(u,u) + B(u,v), m, nmin)
        P7 ← Strass(A(u,v) − A(v,v), B(v,u) + B(v,v), m, nmin)
        C(u,u) ← P1 + P4 − P5 + P7
        C(u,v) ← P3 + P5
        C(v,u) ← P2 + P4
        C(v,v) ← P1 + P3 − P2 + P6
    end if
    return C
end function
```

Từ thuật toán trên ta có công thức truy hồi

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

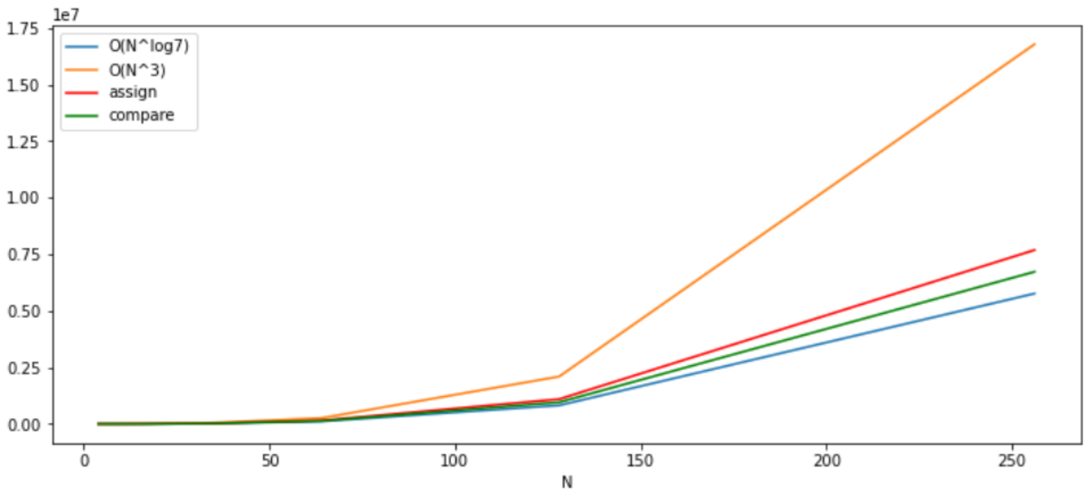
khi đó ta có với $k = log_2 n$

$T(n) = 7T(\overline{a_k a_{k-1}....a_1 0}) + C_2$

$= 7^2 T(\overline{a_k a_{k-1}....a_2 2}) + 7C_2 + C_2$

.

.
.
$= 7^kT(\overline{a_{k2}}) + C_2 \sum_{i=0}^{k-1} 7^i$
$= 7^kT(1) + C_2 \sum_{i=0}^{k-1} 7^i$
$= 7^1C_1 + C_2 \sum_{i=0}^{k-1} 7^i$
$\leq 7^1C_1 + C_2 \sum_{i=0}^{k-1} 7^k$
$= 7^1C_1 + 7^kC_2K$
$= 7^k(C_1 + KC_2)$
$= 7^{\log_2 n}(C_1 + KC_2)$
$= n^{\log_2 n}(C_1 + KC_2)$
$= O(n^{log_2 7}$
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
Bài toán 2 (Bonus)
Có thể nhân 2 ma trận nn 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
Trả lời Không thể nhân 2 ma trận nn với độ phức tạp là $O(n^2)$ vì mỗi ma trận có kích thước nn 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.

```
                Exercise 1
****************************************
>> Matrix A:
 [[ 37 917  36 399]
 [873 174 214 369]
 [672 771 914  45]
 [437 728 587 544]]
>> Matrix B:
 [[400 635 584 831]
 [754 936 533 391]
 [555 659 526 204]
 [569 184 769 732]]

@ Classic Method
>> Result of multiplication between A and B:
 [[ 953229  978947  836136  688706]
 [ 809127  926141  998899 1107261]
 [1383009 1758982 1318760 1079289]
 [1359033 1445832 1370330 1165751]]

@ Strassen Method
>> Result of multiplication between A and B:
 [[ 953229  978947  836136  688706]
 [ 809127  926141  998899 1107261]
 [1383009 1758982 1318760 1079289]
 [1359033 1445832 1370330 1165751]]

@ Compare with result of built-in function:
 [[ 953229  978947  836136  688706]
 [ 809127  926141  998899 1107261]
 [1383009 1758982 1318760 1079289]
 [1359033 1445832 1370330 1165751]]
```