Huấn luyện đội tuyển học sinh giỏi quốc gia tỉnh Long An

Tăng Khải Hạnh

Ngày 28, tháng 10, năm 2017

Mục lục

1 Giới thiệu

Trong các bài toán quy hoạch động trên mảng một chiều, chúng ta có thể gặp một số bài toán yêu cầu tìm các giả trị ở những chỉ số rất lớn. Ví dụ, với dãy số Fibonacci $(F_n)_{n\geq 0}$, ta có thể tính toán trong modulo đến khoảng giá trị F_{10^6} trong thời gian cho phép (khoảng 1 đến 2 giây) bằng cách tính tuần tự từ F_2 đến F_n với các giá trị khởi tạo $F_0 = F_1 = 1$. Nếu ta cần tính các giá trị F_{10^9}, $F_{10^{18}}$ thì kết quả cần đạt bằng cách tính tuần tự sẽ phải tốn vài phút đến vài giờ, tháng, hoặc năm với độ phức tạp $O(n)$.

Do đó, trong bài viết này, chúng ta dùng một phương pháp tính khác bằng cách sử dụng phương pháp "nhân ma trận" trong Đại số Tuyến tính. Mục tiêu đạt được trong bài viết này là cải thiện thời gian chạy, hay giảm độ phức tạp, của việc tính trong các giá trị trong dãy Fibonacci ở những chỉ số cực lớn với độ phức tạp $O(\log n)$. Lưu ý: trong Khoa học máy tính, cụ thể là Cấu trúc Dữ liệu và Giải thuật, thường hiểu $\log n$ là $\log_2 n$ thay vì $\log_{10} n$ như trong Giải tích.

2 Nhân ma trận

Phép nhân hai ma trận

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{bmatrix}$$

và

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,p} \\ b_{2,1} & b_{2,2} & \dots & b_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \dots & b_{n,p} \end{bmatrix}$$

là ma trận

$$AB = \begin{bmatrix} \langle A^T_{1,*}, B_{*,1}\rangle & \langle A^T_{1,*}, B_{*,2}\rangle & \dots & \langle A^T_{1,*}, B_{*,p}\rangle \\ \langle A^T_{2,*}, B_{*,1}\rangle & \langle A^T_{2,*}, B_{*,2}\rangle & \dots & \langle A^T_{2,*}, B_{*,p}\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle A^T_{m,*}, B_{*,1}\rangle & \langle A^T_{m,*}, B_{*,2}\rangle & \dots & \langle A^T_{m,*}, B_{*,p}\rangle \end{bmatrix}.$$

Trong đó, $\langle \cdot, \cdot \rangle$ là tích vô hướng giữa hai vector. $A_{i,*}$ là dòng thứ i của ma trận A và $A^T_{i,*}$ là phép chuyển vị dòng thứ i của ma trận A thành vector cột. Tương tự, $B_{*,i}$ là vector cột thứ i của ma trận B. Lưu ý rằng ma trận A có kích thước là $m \times n$, ma trận B có kích thước là $n \times p$ và ma trận tích AB có kích thước là $m \times p$. Độ phức tạp tính toán là $O(m \times n \times p)$.

Ngoài ra, ta cần lưu ý tính chất kết hợp của phép nhân ma trận. Cụ thể, $ABC = (AB)C = A(BC)$ với A, B, và C là các ma trận. Tính chất này hỗ trợ trong việc tính nhanh lũy thừa của ma trận hiệu quả.

Phép tính nhân ma trận được trình bày trong nhiều sách Đại số Tuyến tính. Có thể tham khảo tại Chương 3 trong sách [1].

3 Lũy thừa ma trận

Trong phần này, ta cần tính lũy thừa của một ma trận vuông A^k với k nguyên và $k \geq 0$. Tương tự với phép tính nhanh lũy thừa của số nguyên và lợi dụng tính kết hợp, ta có thể chia cách tính của A^k thành 2 trường hợp sau:

- *k chẵn.* Trong trường hợp này, ta có thể tính được $A^k = A^{k/2} \times A^{k/2}$. Như vậy ta chỉ cần tính 1 lần duy nhất $A^{k/2}$ và sau đó tiến hành phép nhân ma trận để đạt được A^k như đã trình bày ở phần trước;
- *k lẻ.* Tương tự như trên, ta có thể tính $A^k = A^{\lfloor k/2 \rfloor} \times A^{\lfloor k/2 \rfloor} \times A$.

Như vậy, việc tính nhanh lũy ma trận cần độ phức tạp $O(d^3 \times \log k)$ với $d \times d$ là kích thước của ma trận A.

Phép tính lũy thừa nhanh đối với ma trận dựa trên cách tính lũy thừa nhanh đối vố số nguyên, tham khảo Mục 1.2.2, Chương 1 trong sách [2].

4 Bài toán: Tính toán giá trị phần tử trong dãy số Fibonacci

Dãy số Fibonacci $(F_n)_{n \geq 0}$ được định nghĩa như sau:

- $F_0 = F_1 = 1$,
- $F_n = F_{n-1} + F_{n-2}$ với $n \geq 2$.

Yêu cầu: cho số nguyên n với $n \geq 0$, tìm giá trị $F_n \mod p$ với p là một số nguyên dương cho trước và có kích thước 32 bit.

Để giải bài toán này, ta nhận xét:

- $F_{n-1} = 1 \times F_{n-1}$,
- $F_n = 1 \times F_{n-1} + 1 \times F_{n-2}$.

Dưới dạng ma trận, ta có thể viết như sau

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} F_{n-2} \\ F_{n-1} \end{bmatrix} = \begin{bmatrix} F_{n-1} \\ F_n \end{bmatrix}.$$

Tương tự, ta cũng có

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} F_{n-3} \\ F_{n-2} \end{bmatrix} = \begin{bmatrix} F_{n-2} \\ F_{n-1} \end{bmatrix}.$$

Do đó, ta suy ra được

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^2 \begin{bmatrix} F_{n-3} \\ F_{n-2} \end{bmatrix} = \begin{bmatrix} F_{n-1} \\ F_n \end{bmatrix}.$$

Ta cũng suy ra

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^{n-1} \begin{bmatrix} F_0 \\ F_1 \end{bmatrix} = \begin{bmatrix} F_{n-1} \\ F_n \end{bmatrix}.$$

Như vậy, để tìm số Fibonacci thứ n, ta cần tính

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^{n-1}.$$

Việc tính toán này tốn thời gian $2^3 \times O(\log n) = O(\log n)$ vì 2^3 là hằng số. Để có kết quả, ta cần như ma trận trên với vector

$$\begin{bmatrix} F_0 \\ F_1 \end{bmatrix}$$

và điều này tốn thời gian $2 \times 2 \times 1 = O(1)$. Do đó, việc tính toán số Fibonacci thứ n cần thời gian $O(\log n)$.

Phương pháp tính này được tham khảo tại [3].

5 Bài toán: tổng bình phương các số Fibonacci

Với $(F_n)_{n \geq 0}$ là dãy Fibonacci. Yêu cầu: tính $F_1^2 + F_2^2 + \ldots + F_n^2$ theo modulo p với p là số nguyên 32 bit.

Đặt $S_n = F_1^2 + F_2^2 + .. + F_n^2$. Ta có các nhận xét sau đây:

- $F_{n-1} = F_{n-1}$,
- $F_n = F_{n-1} + F_{n-2}$,
- $F_{n-1}^2 = F_{n-1}^2$,
- $F_n^2 = (F_{n-1} + F_{n-2})^2 = F_{n-1}^2 + 2 \times F_{n-1}F_{n-2} + F_{n-2}^2$,
- $S_n = S_{n-1} + F_n^2 = S_{n-1} + F_{n-1}^2 + 2 \times F_{n-1}F_{n-2} + F_{n-2}^2$,
- $F_nF_{n-1} = (F_{n-1} + F_{n-2})F_{n-1} = F_{n-1}^2 + F_{n-1}F_{n-2}$.

Dựa vào các nhận xét trên, ta dựng được biểu thức sau:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_{n-2} \\ F_{n-1} \\ F_{n-2}^2 \\ F_{n-1}^2 \\ S_{n-1} \\ F_{n-1}F_{n-2} \end{bmatrix} = \begin{bmatrix} F_{n-1} \\ F_n \\ F_{n-1}^2 \\ F_n^2 \\ S_n \\ F_nF_{n-1} \end{bmatrix}.$$

Suy luận tương tự như vài tao1n trước, ta cũng tìm được kết quả S_n trong thời gian $O(\log n)$.

Những bài tập về hệ thức đệ quy được sử dụng dựa trên Chương 8 sách [4].

Tài liệu

[1] C. Meyer. *Matrix analysis and applied linear algebra*, volume 2. Siam, 2000.

[2] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill, Inc., 2006.

[3] Ứng dụng phép nhân ma trận trong giải bài tập tin học, http://www.vnschool.net/modules.php?name=News&file=article&sid=3782.

[4] K. Rosen. *Discrete Mathematics and the Applications*. McGraw-Hill, 2012.