

Chuyên đề nâng cao:

Tìm ước của một số bằng kỹ thuật tìm kiếm nhị phân

Tác giả: Phan Thành Hưng (PTNK, ĐHQG TP. HCM)

I. Giới thiệu

Cho bài toán sau:

“Cho một số E , hãy xác định một ước số D của E sao cho $1 < D < E$ ”

- **Lưu ý:**

- Những số x thỏa mãn $1 < x < E$ và E chia hết cho x được gọi là **ước không tầm thường (non-trivial divisor)**.

II. Chú thích

1. Độ phức tạp thời gian

Độ phức tạp thời gian, được viết dưới dạng Big-O. Đây là thước đo tốc độ của thuật toán. Nó cho biết khi số càng lớn thì máy tính sẽ mất thêm bao nhiêu bước tính toán để ra kết quả. Ký hiệu $O(\dots)$ giúp ta phân loại thuật toán nào “nhanh” hay “chậm”.

Có thể nói, $O(f(n))$ nghĩa là số vòng lặp sẽ tỉ lệ thuận với $f(n)$. Khi biểu diễn, ta có thể lược bỏ các hằng số và các thành phần bậc thấp không đáng kể. Ví dụ: $O(3n + 2) = O(n)$, $O(\pi n^2 + n) = O(n^2), \dots$

III. Thuật toán

1. Thuật toán vét cạn

Ta có thể thử lần lượt bằng tay các số nguyên tố: 2, 3, 5, 7, 11, ... và kiểm tra xem E có chia hết hay không. Tuy nhiên, cách này rất chậm và nếu ước nguyên tố của số đó là một số rất lớn thì chắc chắn không thể nào thử hết được.

Còn một cách khác là dùng tính năng FACT trong máy tính, tuy nhiên nếu ước nguyên tố nhỏ nhất của số đó lại quá lớn thì máy tính cũng chịu thua và phải trả ra kết quả trong ngoặc đơn. Ví dụ, khi bạn cố thử phân tích 4056187 ($= 2011 \times 2017$) thì máy tính sẽ trả ra $\boxed{(4056187)}$, hoàn toàn bất lực :<

2. Thuật toán tối ưu

a. Tận dụng hàm tổng và tích

Vì mỗi ước đều phải đi với cặp: $p \times q = E$ nên ta có $\min(p, q) \leq \sqrt{E}$, nghĩa là nếu E là hợp số, thì luôn tồn tại một ước không tầm thường $\leq \sqrt{E}$. Ta có thể dùng một hàm sigma (tổng) để duyệt x từ 2 đến $\sqrt{E}$ và kiểm tra xem, nếu tới giá trị x đầu tiên là ước của E thì x là ước của E , và phần dư E khi chia cho x lúc này bằng 0, nên ta có thể cố tình thực hiện một phép chia cho phần dư đó để tạo ra Math ERROR và dừng vòng lặp. Ta có thể thực hiện như sau:

$$\sum_{x=2}^{\text{Int}(\sqrt{E})} \left(\frac{0}{E - x \text{Int}(E \div x)} \right)$$

Nếu biểu thức báo lỗi **Math ERROR**, nghĩa là trong đoạn $[2; \sqrt{E}]$ tồn tại ít nhất một ước không tầm thường (cần tìm). Tuy nhiên, ta còn có thể sử dụng hàm tích để trả ra kết quả mà không cần báo lỗi như sau:

$$\prod_{x=2}^{\text{Int}(\sqrt{E})} \left(1 - \text{Int} \left(\frac{1}{E - x \cdot \text{Int}(E \div x) + 1} \right) \right)$$

Nếu biểu thức trả ra 1, trong đoạn $[2; \sqrt{E}]$ tồn tại ít nhất một ước, ngược lại nếu biểu thức trả ra 0. Nguyên lý hoạt động của nó dựa trên tính chất của phép AND tương đương với phép nhân trên các giá trị 0/1. Một biểu thức là tích của các giá trị 0 hoặc 1 trả ra kết quả 1 chỉ khi tất cả các thừa số trong biểu thức đó đều là 1. Chỉ cần có ít nhất một giá trị trả ra 0 thì kết quả của biểu thức sẽ là 0 ngay lập tức. Điều đó giống với việc, chỉ cần tìm ra được ít nhất một số x là ước của E trong đoạn $[2; \sqrt{E}]$ thì kết quả sẽ là 0 ngay, còn ngược lại, mọi số đều không phải là ước thì kết quả là 1. Do vậy, nếu $E \bmod x > 0$, nghĩa là x không phải ước của E , ta gán cho nó giá trị là 1. Ngược lại, nếu phần dư đó là 0, ta gán cho nó giá trị 0. Nhiệm vụ của hàm tích ở đây là nhân hết các giá trị đó lại, ta sẽ được kết quả ta cần.

Đặt $\delta_E(x) = (E \bmod x) + 1$, ta có $\delta_E(x) \geq 1$.

Đặt $f(y) = 1 - \text{Int} \left(\frac{1}{y} \right)$, chính là phần biểu thức chính trong hàm tích.

Nếu $\delta_E(x) = 1$, nghĩa là x là ước của E . Ta có $f(\delta_E(x)) = 1 - \text{Int} \left(\frac{1}{\delta_E(x)} \right) = 0$.

Nếu $\delta_E(x) > 1$, nghĩa là $0 \leq \frac{1}{\delta_E(x)} < 1$ nên $f(\delta_E(x)) = 1 - \text{Int} \left(\frac{1}{\delta_E(x)} \right) = 1$.

Dễ thấy, $f(\delta_E(x))$ chính là biểu thức biểu diễn dành cho mỗi x , nếu x là ước của E thì nó sẽ mang giá trị 0, và ngược lại thì nó mang giá trị 1. Ta chỉ cần dùng hàm tích để nhân các giá trị lại với nhau và ra được kết quả cần tìm.

b. Tìm ra vị trí của ước

Từ mục trên, ta có biểu thức kiểm tra xem trong đoạn $[M; \sqrt{E}]$ có tồn tại ít nhất một ước nào của E không.

$$\lambda_E(M) = \prod_{x=M}^{\text{Int}(\sqrt{E})} \left(1 - \text{Int}\left(\frac{1}{E - x \cdot \text{Int}(E \div x) + 1}\right) \right)$$

Biểu thức trên trả ra 0 nếu tồn tại ít nhất một ước của E và 1 nếu không tồn tại.

Dễ thấy, $\lambda_E(M)$ là hàm đơn điệu, vì nếu trong đoạn $[M; \sqrt{E}]$ tồn tại ít nhất một ước, thì trong đoạn $[M-1; \sqrt{E}]$ chắc chắn cũng phải tồn tại ít nhất một ước như vậy, nghĩa là nếu $\lambda_E(M) = 0$ thì $\lambda_E(M-1) = 0$. Tương tự, nếu $\lambda_E(M) = 1$ thì $\lambda_E(M+1) = 1$. Giá trị của $\lambda_E(\forall M)$ sẽ trông như sau: $\dots, 0, 0, 0, 0, \dots, 0, 0, \underbrace{1, 1, \dots, 1, 1, 1, 1, \dots}_{\text{không tồn tại ước}}$

Lúc này, ta hoàn toàn có thể tìm kiếm nhị phân để tìm M lớn nhất mà $\lambda_E(M) = 0$, nghĩa là giá trị M thỏa $\lambda_E(M) = 0$ và $\lambda_E(M+1) = 1$. Nếu trong đoạn $[M; \sqrt{E}]$ tồn tại ít nhất một ước của E , mà đoạn $[M+1; \sqrt{E}]$ lại không tồn tại bất kì ước nào của E , thì buộc M phải là ước của E , chính là ước ta cần tìm.

c. THUẬT TOÁN CHÍNH

Dựa vào lý thuyết trên, ta dễ dàng hình dung được thuật toán như sau:

- ✚ Khởi tạo hai giá trị miền A và B ban đầu lần lượt mang giá trị 2 và $\sqrt{E}$.
- ✚ Tìm giá trị giữa: $M = \text{Int}\left(\frac{A+B+1}{2}\right)$.
- ✚ Nếu $\lambda_E(M) = 0$ ta gán $A := M$ (nhằm thu hẹp miền giá trị lên). Ngược lại, nếu $\lambda_E(M) = 1$, ta gán $B := M-1$ (thu hẹp miền giá trị xuống).

Lặp lại cho tới khi nào $A \geq B$.

Nếu E không chia hết cho A thì E là một số nguyên tố và không tại ước không tầm thường. Còn nếu A là ước của E luôn thì ước không tầm thường (kết quả cần tìm) chính là A .

Số lượng lần để tìm kiếm nhị phân là $O(\log_2 \sqrt{E})$, lượng vòng lặp của λ_E là $O(\sqrt{E})$. Do vậy, độ phức tạp thuật toán là $O(\sqrt{E} \log_2 \sqrt{E})$.



Lưu ý rằng, thuật toán vét cạn bằng hàm tích chỉ đóng vai trò là một “bộ cảm biến” phát hiện sự tồn tại của ước số trên một phạm vi rộng. Để xác định vị trí của một ước cụ thể, ta phải sử dụng thuật toán tìm kiếm nhị phân. Bản chất của tìm kiếm nhị phân ở đây không phải là để thay thế phép duyệt vét cạn, mà là công cụ để xác định vị trí dựa trên kết quả từ phép nhân đó.

➤ Cài đặt vào CASIO

Tận dụng hàm $\text{RanInt}\#(A, B)$ báo lỗi nếu $A \geq B$, ta có thể dùng nó để dừng tự động.

Cài đặt thuật toán hoàn chỉnh vào máy tính Casio fx-580 VN X:

$$A = 0AEMBD \times \text{RanInt}\#(A, \text{Int}(B)) + DA + (1 - D)M:$$

$$B = (1 - D)B + D(M - 1):$$

$$M = \text{Int}\left(\frac{A + B + 1}{2}\right):$$

$$D = \prod_{x=M}^{\text{Int}(\sqrt{E})} \left(1 - \text{Int}\left(\frac{1}{E - x \cdot \text{Int}(E \div x) + 1}\right)\right)$$

Giá trị ban đầu: $A = 0$, $E =$ dữ liệu đầu vào, $M = 2$, $B = \sqrt{E}$, $D = 0$.

Cách dùng: Lặp đa câu lệnh tới khi gặp **Argument ERROR** thì dừng. Kiểm tra lại thêm một bước: nếu E chia hết cho A , kết quả trong biến A . Nếu không, E là số nguyên tố.