

Hàm Mobius

Hàm Mobius

Nguồn: [Quora](#) [🔗](#)

Trong bài viết này mình sẽ giới thiệu với các bạn một chủ đề vô cùng thú vị trong số học - Hàm nghịch đảo Mobius, cũng như cách ứng dụng để giải quyết một số bài toán.

Một lời khuyên dành cho bạn đọc là các bạn nên tự chứng minh những công thức được đề cập để hiểu rõ hơn bản chất bài toán.

Các định nghĩa

Trước khi bắt đầu, các bạn hãy ghi nhớ một số định nghĩa sau đây để việc tiếp thu những kiến thức ở dưới được dễ dàng hơn.

Một hàm $f(n) : N \rightarrow C$ được coi là **hàm nhân tính** ([Multiplicative Function](#) [🔗](#)) nếu với mọi cặp số nguyên tố cùng nhau $n, m \in N$ ta có $f(mn) = f(m)f(n)$

Một số n phân tích ra thừa số nguyên tố sẽ có dạng $n = p_1^{a_1} \cdot p_2^{a_2} \cdot p_3^{a_3} \cdot \dots \cdot p_r^{a_r}$, với p_i là ước nguyên tố của n .

Ký hiệu $a \parallel b$ có nghĩa là a là ước của b , hay b chia hết cho a

Hàm định danh Dirichlet $e(n)$ (Dirichlet identity function):

- $e(n) = 1$ với $n = 1$
- $e(n) = 0$ với $n > 1$

$$I(n) = 1 \text{ với mọi } n \in N$$

$$id(n) = n \text{ với mọi } n \in N$$

Hàm Mobius $\mu(n)$:

- $\mu(1) = 1$
- $\mu(n) = 0$ nếu tồn tại $a_i > 1$
- $\mu(n) = (-1)^r$ nếu $n = p_1 \cdot p_2 \cdot p_3 \cdot \dots \cdot p_r$, hay $a_i = 1$ với mọi i
- Có thể chứng minh được rằng $\mu(n) = \sum_{d \parallel n, d < n} \mu(d)$ với $n > 1$ và tính được $\mu(n)$ bằng cách sử dụng [Sàng](#) [🔗](#) :

```

mu[1] = 1;
for (int i = 1; i <= N; i++)
for (int j = 2*i; j <= N; j += i)
mu[j] -= mu[i];
...

```

Với mỗi $f(n)$, ta gọi hàm tổng $S_f(n)$ là tổng các $f(d)$ với d là ước của n :

$$S_f(n) = \sum_{d|n} f(d)$$

Phi hàm Euler $\phi(n)$ ([Euler totient function](#) ↗): số lượng các số tự nhiên nhỏ hơn hoặc bằng n và nguyên tố cùng nhau với n (hay số lượng các số $d : 1 \leq d \leq n, \gcd(d, n) = 1$). Các bạn cũng có thể sử dụng sàng để tính $\phi(n)$:

```

1 | for (int i = 1; i <= N; i++) phi[i] = i;
2 | for (int i = 2; i <= N; i++)
3 |     if (phi[i] == i)
4 |         for (int j = i; j <= N; j += i)
5 |             phi[j] -= phi[j]/i;

```

Công thức nghịch đảo Mobius (Möbius inversion formula)

Ta định nghĩa Dirichlet Convolution là một phép toán với 2 hàm $f(n)$ và $g(n)$:

$$f \circ g(n) = \sum_{d_1 \cdot d_2 = n} f(d_1) \cdot g(d_2)$$

hay

$$f \circ g(n) = \sum_{d|n} f(d) \cdot g(n/d)$$

Có thể dễ dàng chứng minh phép toán này có tính giao hoán và kết hợp:

$$f \circ g(n) = g \circ f(n)$$

$$(f \circ g) \circ h(n) = f \circ (g \circ h)(n)$$

Ta có nhận xét rằng:

$$S_f(n) = \sum_{d|n} f(d) = \sum_{d|n} f(d) \cdot I(n/d) = f \circ I(n)$$

$$S_\phi(n) = \sum_{d|n} \phi(d) = id(n)$$

$$S_\mu(n) = \sum_{d|n} \mu(d) = I \circ \mu(n) = \mu \circ I(n) = e(n)$$

$$f \circ e(n) = \sum_{d|n} f(n/d) \cdot e(d) = f(n)$$

Từ đó suy ra:

$$S_f \circ \mu(n) = (f \circ I) \circ \mu(n) = f \circ (I \circ \mu)(n) = f \circ e(n) = f(n)$$

hay

$$f(n) = \sum_{d|n} S_f(d) \cdot \mu(n/d) - \text{đây gọi là Công thức nghịch đảo Mobius} \quad \square$$

Ứng dụng

GCDEX

Đề bài [☐](#)

$$\text{Tính } G = \sum_{i=1}^n \sum_{j=i+1}^n \gcd(i, j) \quad (1).$$

Dễ thấy cách tiếp cận đơn giản nhất cho bài toán này là duyệt tất cả các cặp (i, j) . Độ phức tạp của thuật toán này là $O(k \cdot n^2)$, với k là số phép toán tối đa khi tính \gcd . Chúng ta sẽ đi tìm một lời giải tối ưu hơn sử dụng những kiến thức ở trên.

1. Nhận xét rằng $1 \leq \gcd(i, j) \leq n$ với mọi $1 \leq i < j \leq n$.

Như vậy, biểu thức trên có thể viết lại thành

$$G = \sum_{g=1}^n g \cdot \text{cnt}[g] \quad (2). \text{ Với } \text{cnt}[g] \text{ là số lượng cặp } (i, j) \text{ có } \gcd \text{ bằng } g.$$

Công việc tính $\text{cnt}[g]$ thật sự không hề đơn giản. Ta để ý rằng $\gcd(i, j) = g \Leftrightarrow \gcd(i/g, j/g) = 1$, hay nói cách khác, i/g và j/g phải là 2 số nguyên tố cùng nhau.

2. Ta viết lại (2) thành

$$G = \sum_{g=1}^n h(g) \cdot \text{cnt}[g] \text{ với } h(g) = g$$

3. Giờ chúng ta sẽ tìm cách phân tích $h(n)$ thành hàm tổng của hàm $f(n)$ nào đó, tức là $h(n) = S_f(n) = \sum_{d|n} f(d)$.

Ứng dụng công thức nghịch đảo Mobius, bạn có thể tìm được $f(n) = \sum_{d|n} h(d) \cdot \mu(n/d)$. Một kết quả rất đẹp trong bài toán này là $f(n) = \phi(n)$, việc chứng minh mình xin nhường bạn đọc.

Khi đã biết được $h(n)$ và $\mu(n)$, ta có thể tính $f(n)$ bằng sàng như sau:

```
for (int i = 1; i <= N; i++)
for (int j = i; j <= N; j += i)
f[j] += h[i] * mu[j/i];
...
```

1 | Đoạn code trên chạy trong thời gian $O(N \log N)$ vì với mỗi i và

4. Viết lại (2) một lần nữa ta được:

$$G = \sum_{g=1}^n (\sum_{d|g} f(d)) * cnt[g] \quad (3) \Leftrightarrow G = \sum_{d=1}^n f(d) * (\sum_{g:d|g} cnt[g]) \quad (4)$$

5. Đặt $cnt2[d] = \sum_{g:d|g} cnt[g]$. Hàm này có ý nghĩa là số lượng cặp (i, j) có gcd là bội của d . Đến đây mọi việc đã đơn giản hơn rất nhiều. Các bạn chỉ cần tìm số lượng cặp (i, j) mà i và j đều là bội của d . Có n/d bội của d , nên sẽ có $n/d * (n/d - 1)/2$ cặp.

Vậy (4) trở thành $G = \sum_{d=1}^n f(d) * n/d * (n/d - 1)/2$.

Dễ dàng chứng minh là chỉ có \sqrt{n} giá trị n/d nên ta có thể duyệt từng giá trị của n/d và cộng $n/d * (n/d - 1)/2 * \sum_{k:n/k=n/d} f(k)$ vào kết quả. Bằng tổng tiền tố các bạn có thể truy vấn được $\sum_{k:n/k=n/d} f(k)$ trong $O(1)$ và G trong $O(\sqrt{n})$:

```
for (int i = 1; i <= n; i = j + 1) {
    j = n / (n/i); // vị trí j xa nhất mà n/i = n/j
    res += n/i * (n/i - 1) / 2 * (Sf[j] - Sf[i-1]); // Sf[i] = f[1] + f[2] + f[3] + ... + f[i]
}
...
```

Như vậy thuật toán trên có độ phức tạp $O(N \log N + \sqrt{N} * T)$ với T là số test.

Bài toán tổng quát

Bài toán tổng quát hơn của bài toán trên là tính $G = \sum_{i=1}^n \sum_{j=i+1}^n h(gcd(i, j))$ với h (nên) là một hàm nhân tính. Ví dụ muốn tính $G = \sum_{i=1}^n \sum_{j=i+1}^n gcd^3(i, j)$ thì $h(n) = n^3$.

Các bước tính toán gần như giống với bài toán trên.

Coprime Triples

[Đề bài](#) 

Cho dãy số a_1, a_2, \dots, a_n . Tìm số bộ ba (i, j, k) ($1 \leq i < j < k \leq n$) có $gcd(a_i, a_j, a_k) = 1$. $n \leq 10^5$ và $a_i \leq 10^6$

Ta đưa đề bài này về bài toán: tính $G = \sum_{i=1}^n \sum_{j=i+1}^n \sum_{k=j+1}^n e(gcd(a_i, a_j, a_k))$

1. Viết lại biểu thức trên:

$$G = \sum_g^{maxA} h(g) * cnt[g], \text{ ở đây } cnt[g] \text{ là số lượng bộ ba } (i, j, k) \text{ có } gcd(a_i, a_j, a_k) = 1. maxA = 10^6$$

2. Tìm $f(n)$ bằng công thức nghịch đảo Mobius.






Ở bài toán này $f(n)$ chính bằng $\mu(n)$, việc chứng minh mình cũng xin nhường lại cho bạn đọc.


$$\text{Lúc này } G = \sum_{g=1}^{maxA} (\sum_{d|g} f(d)) * cnt[g]. \Leftrightarrow G = \sum_{d=1}^{maxA} f(d) * cnt2[g].$$

3. Tính $cnt2[d]$ là số bộ ba (i, j, k) có $gcd(a_i, a_j, a_k)$ là bội của d .

4. Duyệt lại d và tính G .

Một số bài luyện tập

- <https://www.codechef.com/JULY10/problems/LCM/> 
- <http://www.codechef.com/COOK29/problems/EXGCD> 
- <https://www.hackerrank.com/challenges/hyperrectangle-gcd> 
- <http://www.spoj.com/problems/LCMSUM/> 
- <http://www.spoj.com/problems/GCDEX2/> 

Nếu các bạn muốn tìm hiểu sâu và đầy đủ hơn về các bài toán liên quan thì có thể tham khảo <http://mathcircle.berkeley.edu/original/Multiplicative.pdf> 

Được cung cấp bởi [Wiki.js](#)