

Hàm nhân tính

Hàm nhân tính

Tác giả: RR

Trong bài viết này, mình sẽ giới thiệu về hàm nhân tính cũng như ứng dụng của nó trong Competitive Programming (lập trình thi đấu).

1. Định nghĩa

Một hàm $f(n)$, $n \in \mathbb{N}$ được coi là **hàm nhân tính** (Multiplicative Function [↗](#)) nếu: Với mọi cặp số nguyên tố cùng nhau $n, m \in \mathbb{N}$ ta có $f(mn) = f(m)f(n)$.

Ví dụ

Xét hàm $f(n)$ là số ước của n . Ta có:

1	$f(1) = 1$
2	$f(2) = 2$
3	$f(3) = 2$
4	$f(4) = 3$
5	$f(5) = 2$
6	$f(6) = 4 = f(2) * f(3)$
7	$f(7) = 2$
8	$f(8) = 4$
9	$f(9) = 3$
10	$f(10) = 4 = f(2) * f(5)$
11	$f(12) = 6 = f(3) * f(4)$
12	$f(18) = 6 = f(2) * f(9)$

Ta có thể chứng minh hàm f là hàm nhân tính như sau:

- Xét m và n nguyên tố cùng nhau.
- Để chứng minh $f(m * n) = f(m) * f(n)$, ta cần chứng minh 2 chiều:
 1. Mỗi ước của $m * n$ ứng với 1 cặp (a, b) với a là ước của m và b là ước của n .
 - Điều này đúng do mỗi ước của $m * n$ ta luôn tách được thành tích của 2 số $a * b$ với a là ước của m và b là ước của n .
 2. Mỗi cặp (a, b) với a là ước của m và b là ước của n ứng với 1 ước của $m * n$.

- Điều này đúng do với mỗi cặp (a, b) thì $a \setminus * b$ là ước của $m \setminus * n$.

Như vậy, $f(m \setminus * n) = f(m) \setminus * f(n)$.

2. Ứng dụng 1

Giờ ta xét bài toán sau:



Cho số N không quá 10^5 . Tính tất cả các $f(i)$ với $i \leq N$.

Để làm những bài dạng này, ta sẽ có 3 bước:

1. Chứng minh f là hàm nhân tính.
2. Tìm công thức cho $f(p^k)$ với p là số nguyên tố.
3. Dùng sàng để tính f trong $O(N * \log N)$.

Nếu bạn chưa biết sàng có thể đọc [ở đây](#).

2.1. Bước 1

Như đã chứng minh ở phần trước, f là hàm nhân tính.

2.2. Bước 2

Với một số nguyên tố p , ta có $f(p^k) = k + 1$, do các ước của p^k là $1, p, p^2, \dots, p^k$.

2.2. Bước 3

Đầu tiên, ta dùng sàng để:

1. Biết được số nào là số nguyên tố.
2. Với mỗi số không phải nguyên tố, tìm được 1 ước nguyên tố.
3. Từ 2, ta nhanh chóng kiểm tra được 1 số có dạng p^k hay không.

Như vậy, ta có thể cài đặt như sau:

```
const int MN = 1e6 + 11;

int sieve[MN];           // Sàng số nguyên tố. Sau khi sàng:
                          // - sieve[i] = 0 nếu i là số nguyên tố
                          // - ngược lại sieve[i] = một ước bất kỳ
pair<int, int> pk[MN];     // Nếu i có dạng p^k, pk[i] = {p, k}.
                          // Ngược lại, pk[i] = {-1, 0}
int ndiv[MN];            // ndiv[i] = Số ước của i.
```

```

11  int main() {
12      // Sàng số nguyên tố
13      for (int i = 2; i <= 1000; i++) // số không nguyên tố có 1
14          if (!sieve[i]) {
15              for (int j = i*i; j <= 1000000; j += i)
16                  sieve[j] = i;
17          }
18
19      ndiv[1] = 1;
20
21      for (int i = 2; i <= 1000000; i++) {
22          if (!sieve[i]) {
23              // i là số nguyên tố.
24              pk[i] = make_pair(i, 1);
25              ndiv[i] = 2;
26          }
27          else {
28              int p = sieve[i]; // p là ước bất kỳ của i.
29
30              if (pk[i/p].first == p) { // i = p^k
31                  pk[i] = make_pair(p, pk[i/p].second + 1);
32                  ndiv[i] = pk[i].second + 1; // ndiv[p^k] = k+1
33              }
34              else {
35                  pk[i] = make_pair(-1, 0);
36                  // Phân tích i = u*v, với gcd(u, v) = 1.
37                  int u = i, v = 1;
38                  while (u % p == 0) {
39                      u /= p;
40                      v = v * p;
41                  }
42                  ndiv[i] = ndiv[u] * ndiv[v];
43              }
44          }
45      }
46  }

```

3. Ứng dụng 2

Ta xét bài toán sau:



Cho số N không quá 10^{12} . Tính $f(N)$

Chú ý ở bài toán trước ta cần tính nhiều giá trị của $f(N)$ với N nhỏ, còn trong bài này ta chỉ cần tính duy nhất 1 giá trị của $f(N)$ với N lớn.

Cũng như trên, ta sẽ làm theo 3 bước chính:

1. Chứng minh f là hàm nhân tính.
2. Tìm công thức cho $f(p^k)$ với p là số nguyên tố.
3. Phân tích N thành thừa số nguyên tố để tính $f(N)$ trong $O(\sqrt{N})$.

Vì 2 bước đầu giống hệt phần trước nên mình sẽ không nhắc lại.

Ở bước 3, bạn chỉ cần xét tất cả các số từ 1 đến \sqrt{N} , từ đó phân tích được thành thừa số nguyên tố. Code như sau:

```

1  int n;
2  int res = 1; // kết quả
3  for (int i = 2; i*i <= n; i++) {
4      if (n % i == 0) {
5          // i là ước nguyên tố của n
6          // (nếu i không nguyên tố, và có ước p, thì ở bước trước đó
7          // ta đã chia n cho p đến khi n không chia hết cho p).
8          int u = 1, k = 0;
9          // u = i^k là lũy thừa lớn nhất của i mà là ước của n.
10         while (n % i == 0) {
11             n /= i;
12             u = u * i;
13             k += 1;
14         }
15         res = res * (k + 1);
16     }
17 }
18
19 if (n > 1) { // giá trị hiện tại của n là số nguyên tố
20     res = res * 2;
21 }

```

4. Dirichlet Convolution

Việc chứng minh trực tiếp một hàm là hàm nhân tính như ví dụ về hàm số ước ở trên không hề đơn giản. Chẳng hạn, bạn hãy thử chứng minh hàm $f(n)$ là hàm nhân tính với $f(n)$ là tổng các ước của số n . Dĩ nhiên bạn có thể chứng minh trâu bò bằng cách viết ra một đồng công thức, tuy nhiên ở mục này mình sẽ hướng dẫn các bạn một phương pháp kỳ diệu hơn.

Với 2 hàm f và g là hàm nhân tính, ta có một hàm nhân tính mới $f \times g$:

$$(f \times g)(n) = \sum_{d_1 * d_2 = n} f(d_1) * g(d_2)$$

Một cách biểu diễn khác là:

$$(f \times g)(n) = \sum_{d|n} f(d) * g(n/d)$$

Các bạn chú ý kí hiệu $d||n$ nghĩa là n chia hết cho d .

Chứng minh

Xét a và b nguyên tố cùng nhau. Mỗi ước d của ab có thể phân tích duy nhất dưới dạng $d = rs$ trong đó $r||a$ và $s||b$, do $\gcd(a, b) = 1$.

Do đó:

$$\begin{aligned} (f \times g)(ab) &= \sum_{r||a, s||b} f(rs)g(ab/rs) \\ &= \sum_{r||a, s||b} f(r)f(s)g(a/r)g(b/s) \\ &= \sum_{r||a} f(r)g(a/r) \sum_{s||b} f(s)g(b/s) \\ &= (f \times g)(a)(f \times g)(b) \end{aligned}$$

Như vậy, $(f \times g)$ cũng là hàm nhân tính.

Để hiểu thêm về Dirichlet Convolution, ta xét vài ví dụ:

4.1. Ví dụ 1

Xét hàm $f(n) = 1$ và $g(n) = 1$. Rõ ràng f và g đều là hàm nhân tính.

$$(f \times g)(n) = \sum_{d|n} f(d) * g(n/d) = \sum_{d|n} 1$$

Như vậy $(f * g)(n)$ là số ước của số n và là hàm nhân tính.

4.2. Ví dụ 2

Xét hàm $f(n) = n$ và $g(n) = 1$. Rõ ràng f và g đều là hàm nhân tính.

$$(f \times g)(n) = \sum_{d|n} f(d) * g(n/d) = \sum_{d|n} d$$

Như vậy $(f \times g)(n)$ là tổng các ước của n và là hàm nhân tính.

Tổng quát hơn, với hằng số k bất kỳ, hàm $f(n) = \sum_{d|n} d^k$ là hàm nhân tính.

4.3. Các hàm nhân tính thường gặp

Sau đây là các hàm nhân tính thường gặp. Bạn có thể thử chứng minh những hàm này là hàm nhân tính dựa theo định nghĩa hoặc Dirichlet Convolution. Việc nắm được những hàm này sẽ giúp thuận lợi hơn trong việc giải những bài liên quan đến hàm nhân tính.

- $I(n) = 1$, hàm f luôn bằng 1 với tất cả các giá trị của n .

- $id(n) = n$
- $id_k(n) = n^k$
- $gcd(n, k)$ với k là hằng số
- $\phi(n)$: số số nguyên tố cùng nhau với n và nhỏ hơn hoặc bằng n (Phi hàm Euler)
- $mobius(n)$: Hàm mobius thường được dùng trong các bài toán đếm sử dụng nguyên lý bù trừ:
 - Nếu n có ước là số chính phương khác 1, $mobius(n) = 0$.
 - Nếu n có lẻ ước nguyên tố, $mobius(n) = -1$.
 - Nếu n có chẵn ước nguyên tố, $mobius(n) = 1$. Chú ý $mobius(1) = 1$.
- $f(n) = \sum_{d|n} d$ là tổng các ước của n .
- $f_k(n) = \sum_{d|n} d^k$ với k là hằng số.

5. Tổng kết

Như vậy, nếu bạn chứng minh được một hàm f là hàm nhân tính, và tìm được công thức $O(1)$ cho $f(p^k)$ thì sẽ dễ dàng tính được tất cả các giá trị $f(i), i \leq N$ trong $O(N * \log N)$.

Bạn cũng nên nắm được những hàm nhân tính thường gặp, từ đó giúp nhận dạng bài toán dễ dàng hơn.

Bài luyện tập

- [VNOJ - GCDSUM](#) 

Gợi ý: Xét $f(n) = \sum_{1 \leq i \leq n} gcd(n, i)$.

- [CF #391 - E](#) 

Gợi ý: Tìm cách bỏ phép cộng trong công thức của $f_r(n)$.

- [CF #400 - E](#) 

Reference

- [Wikipedia](#) 

Được cung cấp bởi [Wiki.js](#)