數論 II

1

前言

之前大家學過了一些數論，這次再繼續精進我們的數論技巧吧！這裡會探討比較進階的話題，像是各種數論函數、莫比烏斯反演、轉換等。讓我們來進入這個神奇的世界吧！

數論函數

何謂數論函數

顧名思義，數論函數就是「數論常常探討、用到的函數」！數論函數通常為 $\mathbb{Z} \to \mathbb{Z}$（整數打到整數）的數字，所以如果沒有註明，就滿足之。這個章節所指的「數」指「整數」。我們主要想要探討的是「乘法函數」：

定義 1.0.1: 乘法函數

一個函數 $f(x)$ 為一個**乘法函數**，若其滿足對於兩個互質的數字 a、b，

$$f(ab) = f(a)f(b)$$

若 a、b 不需要互質，則其被稱為一個**完全乘法函數**。一些乘法函數的例子包括：$f(x) = 0$（0 是任意數字）、$f(x) = 1$、$f(x) = x$、$f(x) = x^k$（k 是任意數字）等。

歐拉 φ 函數

定義 1.0.2: 歐拉 φ 函數

對於一個正整數 n，我們定義

$$\varphi(n) = \#\{x \mid (n, x) = 1\}$$

也就是「小於 n 的正整數中，與 n 互質的個數」。

φ 是乘法函數

那要如何證明其為乘法函數呢？假設 a, b 互質。則我們將所有少於 ab 的數字分為三個集合：X, Y, Z，分別代表與 a 互質、與 b 互質、與 ab 互質。令 $x \in X$、$y \in Y$，則存在一個數字 $t \in Z$ 滿足

$$\begin{cases} t \equiv x \mod a \\ t \equiv y \mod b \end{cases}$$

也就是說，對於每一組 (x, y)，都可以找到唯一一個相對應的 t；對於每一個 t，也都可以找到唯一一組相對應的 (x, y)。則由中國剩餘定理，$t = (Ax + By) \pmod{ab}$，$A = b \cdot [b^{-1} \pmod{a}]$，$B = a \cdot [a^{-1} \pmod{b}]$。現在假設存在一組 $x' \in X$、$y' \in Y$，且 $t' = (Ax' + By') = t$。我們想要證明 $x' = x$ 且 $y' = y$（也就是說的確是唯一的）

首先，可以知道 $Ax + By = Ax' + By'$。故 $A(x - x') + B(y - y\text{'}) = 0$。兩邊取 a 的餘數：

$$(x - x') + 0 \equiv 0 \pmod{a}$$

故 $x \equiv x' \pmod{a}$，但是因為 x, x' 皆小於 a，故 $x = x'$。同理，$y = y'$。

n 為質數冪

不難看出，若 p 為質數，則 $\varphi(p) = p - 1$。那如果是質數的冪次呢？在小於 p^k 的數字中，唯有是 p 的倍數的數字不符合。這種數字有 $\frac{p^k}{p} = p^{k-1}$ 個，扣除則可以得到：

$$\varphi(p^k) = p^k - p^{k-1}$$

任意數的 φ 值

有了是乘法函數的性質，且有質數冪了，就可以計算任何數的 φ 值了！假設我們要計算 $\varphi(n)$ 值。首先，先來質因數分解：

$$n = \prod_{i=1}^{\infty} p_i^{\alpha_i}$$

此處 p_i 為第 i 個質數，α_i 為第 i 個質數的次方。則根據以上，

$$\varphi(n) = \prod_{i=1}^{\infty} \varphi(p_i^{\alpha_i}) = \prod_{i=1}^{\infty} \left[p_i^{\alpha_i} - p_i^{\alpha_i - 1}\right]$$

如果將 n 提出來，則會有更漂亮的公式：

$$\varphi(n) = n \cdot \prod_{p|n} (1 - \frac{1}{p})$$

且 p 為質數。

1.1 要怎麼計算 φ 值呢？

可以在之前學過的線性篩法做一些修改——

```
const int maxN = 1e5;

int phi[maxN];
bool isPrime[maxN];
vector<int> primes;

void sieve(){
    fill(isPrime, isPrime + maxN, true);
    for(int i = 2; i < maxN; i++){
        if(isPrime[i]){
            phi[i] = i - 1;
            primes.push_back(i);
        }
        for(int p : primes){
            if(p*i >= maxN) break;
            isPrime[p*i] = false;
            if(i % p) phi[i*p] = phi[i]*(p - 1);
            else {
                phi[i*p] = phi[i]*p;
                break;
            }
        }
    }
}
```

就可以在 $O(n)$ 的時間內得到 n 以下所有數字的 φ 值了！

μ 函數

要介紹的第二個函數就是 μ 函數了！雖然定義可能顯得有點彆扭，但是其實它是非常有用的！

定義 1.1.1: μ 函數

對於一個正整數 n，我們定義 $\mu(n)$ 為：

$$\begin{cases} 0, & n \text{ 有平方因數} \\ 1, & n \text{ 有偶數個質因數} \\ -1, & n \text{ 有奇數個質因數} \end{cases}$$

要怎麼計算 μ 值呢？

其實也是老套路，只要修改線性曬法就好了，甚至也可以在一次質數篩中，同時算完 φ、μ 的值！利用其積性即可：

```cpp
const int maxN = 1e5;

int mu[maxN];
bool isPrime[maxN];
vector<int> primes;

void sieve(){
    fill(isPrime, isPrime + maxN, true);
    for(int i = 2; i < maxN; i++){
        if(isPrime[i]){
            mu[i] = -1;
            primes.push_back(i);
        }
        for(int p : primes){
            if(p*i >= maxN) break;
            isPrime[p*i] = false;
            if(i % p) phi[i*p] = -phi[i];
            else {
                phi[i*p] = 0;
                break;
            }
        }
    }
}
```

這個函數是乘法函數的性質很顯然，就不證明了！那這個看起來很彆扭的東西到底有用在哪裏呢？當然就是——莫比烏斯反演啦！

1.2 莫比烏斯反演

終於來到了我們的重頭戲——莫比烏斯反演！名字聽起來很神秘的莫比烏斯反演，讓我們來揭開其神秘的面目吧！

定義 1.2.1: 狄利克雷卷積

對於兩個函數 f 和 g，我們定義他們的卷積 $f \star g$ 為：

$$(f \star g)(n) = \sum_{d|n} f(d)g(\frac{n}{d})$$