數論 I

1

數論主要是在探討整數的性質，特別是整數方程式、整數之間的整除關係與同餘關係等。

1.1 質數

什麼？你都進高中了還不知道什麼是質數？

定義 1.1.1: 質數

恰好有兩個正因數，即 1 和自身的正整數稱為質數

一個合數 N 的最小質因數必定不大於 $\sqrt{N}$，因此我們可以從 $2, 3, \ldots, \sqrt{N}$ 一一檢查是否有整除來檢驗一個數字是否為質數或者對這個數字質因數分解，複雜度 $O(\sqrt{N})$。

質數篩法

當我們要尋找一個範圍內的質數時，質數篩法就派上用場了。與其暴力枚舉一個一個尋找因數，不如枚舉小數字的倍數。由小到大開始枚舉 2 的倍數，3 的倍數，4 的倍數，$\ldots$，N 的倍數便可以得知有哪些合數，複雜度 $\frac{N}{2} + \frac{N}{3} + \frac{N}{4} + \cdots + \frac{N}{N} = N \cdot \sum_{k=2}^{N} \frac{1}{k} = O(N \log N)$。有幾個小優化：只針對質數進行枚舉、枚舉到 k 的時候只枚舉不小於 k^2 的倍數（因為合數 N 最小的質因數不大於 $\sqrt{N}$），複雜度進化到 $O(N \log \log N)$，而且常數超小，似乎叫做埃式篩。另外有 $O(N)$ 的歐拉線性篩的方法不過這邊略過。

```cpp
bool isPrime[N];
void sieve(int n) {
    for(int i = 2; i <= n; i++) isPrime[i] = true;
    for(int i = 2; i*i <= n; i++) if(isPrime[i]) {
        for(int j = i*i; j <= n; j += i) {
            isPrime[j] = false;
        }
    }
}
```

質因數分解

每一個正整數 N 可以唯一分解為

$$N = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$$

其中 p_i 為相異質數。
N 的正因數個數為

$$(\alpha_1 + 1)(\alpha_2 + 1) \cdots (\alpha_k + 1)$$

N 的正因數和為

$$(1 + p_1 + \cdots + p_1^{\alpha_1})(1 + p_2 + \cdots + p_2^{\alpha_2}) \cdots (1 + p_k + \cdots + p_k^{\alpha_k})$$

怪怪的定理

- （威爾遜定理）p 是質數若且惟若 $(p-1)! \equiv -1 \pmod{p}$
- （費馬小定理）若 p 是質數則 $a \neq 0 \Rightarrow a^{p-1} \equiv 1 \pmod{p}$
- （質數定理）在 N 以下的質數數量大約是 $\dfrac{N}{\ln N}$

1.2 最大公因數

先來個定義

定義 1.2.1: 最大公因數

同時整除 a, b 且最大的 d 被稱為 a, b 的最大公因數，這裡記做 $\gcd(a, b)$

- $\gcd(a, b) = \gcd(b, a), \gcd(a, \gcd(b, c)) = \gcd(\gcd(a, b), c)$（交換律、結合律）
- $\gcd(a, 0) = |a|$（單位元）
- $\gcd(a, b) = \gcd(a, b \pm a) = \gcd(a, b \mod a)$
- $\forall a, b \in \mathbb{Z}, \exists x, y \in \mathbb{Z}, s.t. ax + by = \gcd(a, b)$（Bezout's Theorem）

如果假定除法是 $O(1)$ 的話，由 $\gcd(a, b) = \gcd(b, a \mod b)$ 可以 $O(\log C)$ 求得兩數的最大公因數。不失一般性假設 $a > b$，因為 $a \mod b < b$，所以 $a \geq b + (a \mod b) > 2 \cdot (a \mod b)$，也就是說每做一次有效的模，$a$ 的值就會砍半一次，最多砍半 $O(\log(a) + \log(b))$ 次。通常可以直接用 gcc 的`__gcd`，但是負數要好好處理。

擴展歐幾里得

考慮形如 $ax+by=c$ 的等式，若 $\gcd(a,b)\nmid c$ 顯然沒有解，而若要找出一組 (x,y) 可以用和輾轉相除法類似的形式求解。

假設現在已經算出 $bx'+(a \mod b)y'=d$ 的一組解 (x',y')，改寫成 $bx'+(a-b\cdot\lfloor\frac{a}{b}\rfloor)y'=d$，可以整理為 $ay'+b(x'-\lfloor\frac{a}{b}\rfloor\cdot y')=d$，遞迴 $O(\log C)$ 找到一組解。

```
pair<int,int> extgcd(int a, int b) {
    if(!b) return {1, 0}; // base case
    auto [x, y] = extgcd(b, a%b);
    return {y, x - a/b * y};
}
```

注意得到的不是唯一解，通解為 $(x,y)=(x+\frac{b}{g}t, y-\frac{a}{g}t)$，其中 $t\in\mathbb{Z}, g=\gcd(a,b)$。

1.3 同餘

若兩個整數 a,b 滿足 $m|a-b$，則我們稱他們在模 m 下同餘，又記做 $a\equiv b \pmod m$，C++ 中可以用 % 運算子來取餘數。

算數性質

- $a\equiv c \pmod m, b\equiv d \pmod m \Rightarrow a\pm b\equiv c\pm d \pmod m$
- $a\equiv c \pmod m, b\equiv d \pmod m \Rightarrow ab\equiv cd \pmod m$
- $\forall k\in\mathbb{N}\cup\{0\}, a\equiv b \pmod m \Rightarrow a^k\equiv b^k \pmod m$

通常來說，我們會存界於 $[0,m)$ 的數當作代表，如果有乘法的話請記得考慮要不要使用`long long`

模逆元

大家應該注意到了，上面的算術性質少了加減乘除中的除法！在同餘的世界裡我們通常會以模逆元取代之。若整數 x 滿足 $ax\equiv 1 \pmod m$，則稱 x 是 a 在模 m 下的模逆元，記做 a^{-1}。模逆元讓我們可以使用乘法的等量公理，或者想成直接移項。

$$ax\equiv b \pmod m \Rightarrow x\equiv a^{-1}b \pmod m$$

事實上，存在 a^{-1} 若且惟若 $\gcd(a,m)=1$，但是要怎麼快速求模逆元呢？這邊提供三種方法

- 用擴展歐幾里得求 $ax+m(-y)=1$ 的一組解。
- 如果 m 是質數，利用費馬小定理，$a^{p-1}\equiv 1 \pmod p \Rightarrow a^{-1}\equiv a^{p-2} \pmod p$。
- 如果 m 是質數，利用 DP 的方式，
  $m\%a=m-a\cdot\lfloor\frac{m}{a}\rfloor \Rightarrow a^{-1}\equiv (m\%a)^{-1}\cdot(-\lfloor\frac{m}{a}\rfloor) \pmod m$

當然不論是何種方法，0、與 m 不互質的數都沒有模逆元。

對於第二種方法，p 如果很大怎麼辦？

跟著蕭電這樣做

插播一下快速冪：如何計算 $a^n \pmod m$，其中 n 是一個很大的正整數？筆者還是菜雞的時候曾經想要用常數優化過 ZJ d636，不過當然失敗 XD
考慮分治法，利用 $a^{\left\lfloor \frac{n}{2} \right\rfloor}$ 和 a 本身可以湊出 a^n。另一種思路是依序求出 $a^1, a^2, a^4, a^8, \cdots, a^{2^k}$ 之後，把 n 按照二進位拆成 2 的冪次選出需要的乘起來。

這樣就能快速刻出質數下的模逆元啦

```
long long modpow(long long a, long long n, long long m) {
    long long res = 1;
    while(n) {
        if(n&1) res = res*a%m;
        a = a*a%m;
        n >>= 1;
    }
    return res;
}
long long modinv(long long x, long long p) {
    return modpow(x, p-2, p);
}
```

離散對數

在實數的世界中，$\log_a b = x$ 表示 $a^x = b$，而在同餘的世界中我們偶爾也需要解決這類問題：$a^x \equiv b \pmod p$。

直接枚舉 x 會是 $O(p)$，我們可以利用 meet in the middle 的方法來加速。假設 k 是一個常數，$x = kq + r$，我們把所有 $a^k, a^{2k}, \ldots, a^{qk}$ 丟進一個 hash table 裡面，然後枚舉 r，可以知道 $a^{kq} = y \cdot a^{-r}$，如果在 hash table 裡面找得到對應的 q 就能求得 x 了。如果改假設 $x = kq - r$，就能夠避免求取模逆元，即 $a^{kq} = y \cdot a^r$，複雜度為 $O(\frac{p}{k} + k)$，取 $k = \sqrt{p}$ 能夠把複雜度壓到 $O(\sqrt{p})$。

1.4 例題

做題目的啦

習題 1.4.1: 分組編隊（TIOJ 1668）

求 $[L, R]$ 中質數的個數。$1 \le L \le R \le 2^{31} - 1, R - L \le 2 \times 10^5$

習題 1.4.2: 完全子圖（TIOJ 1459）

有 N 條等式 $x \equiv r_k \pmod{m}_k$，求出 x 最小正整數解或判斷無解。
（中國剩餘定理裸題，可以試著用擴展歐幾里得做）

習題 1.4.3: 互質任務（TIOJ 1069）

給定長度 N 且每個數字介於 0 到 9 的序列，找出最長的子序列使得其代表的十進位數字和 M 互質。$N \leq 10^3, M \leq 10^4$

習題 1.4.4: !!!!!!!!!!!!!!!!!!!（TIOJ 1350）

質因數分解 $n!$，有 q 筆測資。$n, q \leq 10000$

習題 1.4.5: Edgy Trees（CF 1139C）

給一棵樹，上面有一些邊是黑色的。在所有不同的長度為 k 的序列 $[a_1, a_2, \ldots, a_k]$ 當中，如果任何 $i < k$ 滿足 a_i 到 a_{i+1} 的簡單路徑上有黑色邊則這個序列是好的。求所有好的序列模 $10^9 + 7$ 的餘數。