数论

GCD最大公约数

ExGCD扩展GCD

求解$ax+by=gcd(a,b)$

$$ax_1+by_1=gcd(a,b) \tag{1}$$

$$bx_2+(a \bmod b)y_2=gcd(b,a \bmod b) \tag{2}$$

$$gcd(a,b)=gcd(b,a \bmod b)$$

$$ax_1+by_2=bx_2+(a-\lfloor\frac{a}{b}\rfloor\times b)y_2$$

$$ax_1+by_2=ay_2+b(x_2-\lfloor\frac{a}{b}\rfloor y_2)$$

$$\begin{cases} x_1=y_2 \\ y_1=x_1-\lfloor\frac{a}{b}\rfloor y_2 \end{cases}$$

```
def exgcd(a, b):
    if b == 0:
        return a, 1, 0
    d, x, y = exgcd(b, a % b)
    return d, y, x - (a // b) * y
```

欧拉函数

> 定义:$\varphi(n)$表示小于等于n并且和n互质的个数

- $\varphi(1)=1$
- 积性函数，若$gcd(a,b)=1$，那么$\varphi(a\times b)=\varphi(a)\times\varphi(b)$
- n为奇数时，$\varphi(2n)=\varphi(n)$

欧拉定理

若$gcd(a,m)=1$,则$a^{\varphi(m)}\equiv 1(\bmod m)$

若m为质数时，$\varphi(m) = m - 1$,可以得到费马小定理

> 扩展欧拉定理

$$a^b \equiv \begin{cases} a^{b \bmod \varphi(m)}, & gcd(a,m) = 1 \\ a^b, & gcd(a,m) \neq 1, b < \varphi(m) \\ a^{(b \bmod \varphi(m)) + \varphi(m)}, & gcd(a,m) \neq 1, b \geq \varphi(m) \end{cases}$$

Lucas定理

lucas定理

$$\binom{n}{m} \bmod p = \binom{\lfloor \frac{n}{p} \rfloor}{\lfloor \frac{m}{p} \rfloor} \times \binom{n \bmod p}{m \bmod p} \bmod p$$

- 要求p较小，不超过1e5

中国剩余定理

$$\begin{cases} x \equiv a1 (\bmod m1) \\ x \equiv a2 (\bmod m2) \\ \ldots \\ x \equiv ak (\bmod mk) \end{cases}$$

求一个最小的正整数x满足以上所有的线性同余方程(m1，m2，mi互质)

设$M = \prod_{i=1}^{k}$ $\qquad M_i = \frac{M}{m_i}$ $\qquad M_i t_i \equiv 1 (\bmod m_i)$

构造一个解$x = \sum_{i=1}^{k} a_i M_i t_i$

证明如下：

- 当$i \neq j$时, $a_j M_j t_j \equiv 0 (\bmod m_i)$
  - 因为M包含m_i，$M_j = \frac{M}{m_j}$
- 当$i = j$时，$a_i M_i t_i \equiv a_i (\bmod m_i)$

所以$x = \sum_{i=2} k a_i M_i t_i (\bmod mi)$满足题意

所以首先求M，然后求M_i和$t_i (M_i^{-1})$

```
def exgcd(a, b):
    if b == 0: return a, 1, 0
```

```python
    d, x, y = exgcd(b, a % b)
    return d, y, x - a // b * y


def CRT(k, a, p):
    n, res = 1, 0
    for i in p: n *= i
    for i in range(k):
        m = n // p[i]
        d, x, y = exgcd(m, p[i])
        res = (res + a[i] * m * x % n) % n
    return (res + n) % n
```

```cpp
void exgcd(LL a,LL b,LL &x,LL &y){
    if(!b){x = 1,y = 0;return;}
    exgcd(b,a % b,y,x);
    y -= a/b*x;
}

LL CRT(int k,LL *a,LL *p){
    LL P = 1,res = 0;
    rep(i,1,k) P *= p[i];
    rep(i,1,k){
        LL m = P / p[i];
        LL x,y;
        exgcd(m,p[i],x,y);
        res = (res + a[i] * m * x % P) % P;
    }
    return (res + P) % P;
}
```