信息学

基础数论三-CRT与BSGS

西南大学附属中学校

信息奥赛教练组

01 线性同余方程组

中国剩余定理(CRT)

在我国古代算书《孙子算经》中，记载着这样一个问题：
“今有物不知其数，三三数之剩二，五五数之剩三，七七数 之剩二，问物几何？”
这个问题通常称为“孙子定理”，民间俗称“韩信点兵”，国 外的书籍上把这个定理叫做“中国剩余定理”。这个问题是世界 数学史上闻名的问题，涉及到数论中一次同余式组的解法，即求 被5除余4，被7除余5，被7除余4的最小正整数。

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \quad\cdots\cdots \\ x \equiv a_n \pmod{m_n} \end{cases}$$

线性同余方程组

也称“孙子定理”

古人发现：
把a_i的变成1，其他的a变成0得到解x
答案就是$\Sigma(x * a_i)$

中国剩余定理(CRT)

对于这样一个m_i之间互质的问题，古人给出了一个巧妙的“构造法”进行求解
用现代数学语言描述如下：

我们令 $m = \prod_{i=1}^{n} m_i$，$M_i = \frac{m}{m_i}$， 设 $t_i \equiv M_i^{-1} (\mod m_i)$。 ($t_i$为$M_i$模$m_i$意义下的逆元，$M_i t_i \equiv 1(\mod m_i)$)

方程组S的通解形式为：$x = a_1 t_1 M_1 + a_2 t_2 M_2 + \cdots + a_n t_n M_n + kM = \sum_{i=1}^{n} a_i t_i M_i + kM$，$k$属于整数

在模M的意义下，方程组(S)只有一个解：$x = \left(\sum_{i=1}^{n} a_i t_i M_i\right) \mod M$

证明如下：
因为 $M_i = \frac{m}{m_i}$ 是除了 m_i 之外所有模数的倍数，所以对于 $\forall k \neq i$，有 $M_i \equiv 0(\mod m_k)$，也即 $a_i M_i t_i \equiv 0(\mod m_k)$。
又因为 $a_i M_i t_i \equiv a_i(\mod m_i)$，所以如果代入 $x = \sum_{i=1}^{n} a_i M_i t_i$，原方程组成立。
可以理解成每个 $a_i M_i t_i$ 只对 i 这个方程有贡献，对其他方程都没有影响。
证毕。

因为$\frac{M}{m_i}$是除m_i之外的所有m的倍数
所以$\forall k \neq i, a_i \frac{M}{m_i} t_i \equiv 0(\mod m_k)$
又有$\frac{M}{m_i} t_i \equiv 1(\mod m_i)$
两边同时乘a_i得$a_i \frac{M}{m_i} t_i \equiv a_i(\mod m_i)$
带入$x = \sum_{i=1}^{k} a_i \frac{M}{m_i} t_i$
原方程组成立
——算法竞赛进阶指南

得到一个解x，这是一个特解
通解表示为x+k*m
求最小整数解，x对m取模即可

代码


```
ll CRT() {
	ll ans = 0;
	ll M = 1;
	ll x, y;
	for (int i = 1; i <= t; i++) M *= m[i]; //累乘mi
	for (int i = 1; i <= t; i++) {
		ll Mi = M / m[i];
		exgcd(Mi, m[i], x, y);
		x = (x % m[i] + m[i]) % m[i]; //最小非负的解
		ans = (ans + mul(a[i] * Mi % M, x , M)) % M;
	}
	return (ans + M) % M; //防止负数
}
```

如果m_i之间不互质怎么办?

扩展中国剩余定理(exCRT)

核心思路：方程两两合并，用扩展欧几里得求解

举个栗子：

$$\begin{cases} x \equiv 2 \pmod 4 \\ x \equiv 4 \pmod 6 \end{cases} \Rightarrow \quad x \equiv 10 \pmod{12}$$

$$\begin{cases} x \equiv 4 \pmod 6 \\ x \equiv 3 \pmod 5 \end{cases} \Rightarrow \quad x \equiv 28 \pmod{30}$$

$$\begin{cases} x \equiv 2 \pmod 4 \\ x \equiv 3 \pmod 6 \end{cases} \Rightarrow \quad \varnothing$$

如下几个特点：

- 新方程与原方程具有同样的形式。
- 新方程的模数，是之前两个模数的lcm.
- 可能存在无解的情况

扩展中国剩余定理(exCRT)

$$\begin{cases} x \equiv a_1 (mod\ m_1) \\ x \equiv a_2 (mod\ m_2) \\ x \equiv a_3 (mod\ m_3) \\ \vdots \\ x \equiv a_n (mod\ m_n) \end{cases}$$

两两合并： $\begin{cases} x \equiv a_1 (mod\ m_1) \\ x \equiv a_2 (mod\ m_2) \end{cases}$ $\xrightarrow{\text{根据同余定义展开}}$ $\begin{cases} x = a_1 + k_1 \times m_1 \\ x = a_2 + k_2 \times m_2 \end{cases} (k_1 \in \mathbb{Z}, k_2 \in \mathbb{Z})$

联立可得 $a_1 + k_1 \times m_1 = a_2 + k_2 \times m_2$

此基础上移项得 $k_1 \times m_1 - k_2 \times m_2 = a_2 - a_1$

设 $A = m_1, B = m_2, C = a_2 - a_1$

则该式可转换为 $A \times k_1 + B \times k_2 = C$

扩展中国剩余定理(exCRT)

$$A \times k_1 + B \times k_2 = C$$

我们可以利用扩展欧几里得算法求解 k_1，然后代入求解，得出一组特解 x_0

$x = x_0 + lcm(m_1, m_2) \times k(k \in \mathbb{Z})$ 等价于 $x \equiv x_0(mod\ lcm(m_1, m_2))$

$$\begin{cases} x \equiv a_1(mod\ m_1) \\ x \equiv a_2(mod\ m_2) \end{cases} \xrightarrow{\text{合并}} x \equiv x_0(mod\ lcm(m_1, m_2))$$

经过n-1次合并，也就是n-1次扩欧后，得到：

通解式为 $x = x_0 + k \times lcm(m_1, m_2, m_3 \cdots, m_n)(k \in \mathbb{Z})$

另一种更数学的证明

前情提示：是一种数学归纳法的思想

假设已经求出前k-1个方程组成的同余方程组的一个解为x

且有$M=\prod_{i-1}^{k-1} m_i$ M=lcm{m_i} (i<=k-1)时也成立

则前k-1个方程的方程组通解为$x+i*M(i\in Z)$

我们就是要求一个正整数t，使得$x+t*M\equiv a_k(\mod m_k)$

转化一下上述式子得$t*M\equiv a_k-x(\mod m_k)$

对于这个式子我们已经可以通过扩展欧几里得求解t

若该同余式无解，则整个方程组无解，若有，则前k个同余式组成的方程组的一个解解为$x_k=x+t*M$

所以整个算法的思路就是求解**k次扩展欧几里得**

参考代码


```
ll excrt()
{
    ll x,y,k;
    ll M=bi[1],ans=ai[1];  //第一个方程的解单独解决
    for(int i=2;i<=n;i++)
    {
        ll a=M,b=bi[i],c=(ai[i]-ans%b+b)%b;//ax≡c(mod b)
        ll gcd=exgcd(a,b,x,y),bg=b/gcd;
        if(c%gcd!=0) return -1; //判断是否无解
         x=((x*c/g)%bg+bg)%bg; //把x转化为最小非负整数解
        //x=mul(x,c/gcd,bg); //避免溢出，建议使用慢速乘代替
        ans+=x*M; //更新前k个方程组的答案
        M*=bg  //M更新，M为前k个方程的lcm
        ans=(ans%M+M)%M;
    }
    return (ans%M+M)%M;
}
```

02 高次同余方程

求解$a^x \equiv b \pmod c$高次同余方程中x的算法
朴素的BSGS只能求解a,c互质时的情况
当a,c互质时，由费马小定理可知，在x大于等于c-1时会出现循环节，若x存在，则x必然小于c-1

做法：
当c很小时，直接枚举0~c-1，检验是否为方程的解
当c很大的时候怎么办呢？

把c分块来做

朴素BSGS

分块思想的考察可以很简单，也可以很难

这里只是朴素的分块思想

分块一般是分成t=ceil(sqrt(c))个，每块里有ceil(sqrt(c))个，这样才能保证t*t>=c

则，x可以表示为x=t*i+j

那么

$a^x \equiv b \pmod c$

代入，$a^{(t*i+j)} \equiv b \pmod c$ （t,i属于正整数，j<t）

移项，$a^{(j)} \equiv a^{-(t*i)} * b \pmod c$

Baby step:枚举j：0~t-1，然后把a^j mod c 放入hash表或map中

Giant step:枚举t*i:t~c-1,算一下$a^{-(t*i)} * b$，然后找一下hash表或map中有没有这个值

若有得到一组解(i,j)，x=t*i+j

扩展BSGS(目前仅作了解)

利用消因数法，a,c使得互质
自行学习：
https://blog.csdn.net/sdau_fangshifeng/article/details/81458934

Thanks

For Your Watching

FZOI
西|大|附 中|信|息|学|竞 赛|
High School Affiliated to Southwest University