同余方程

Quack

2022 年 7 月 18 日

目录

线性同余方程

定义 (线性同余方程)

$$ax \equiv b \pmod{m}$$

对于一个线性同余方程 $ax \equiv b \pmod{m}$，将其转化成二元一次不定方程 $ax + my = b$，然后用之前我们讲过的方法去解即可。时间复杂度 $O(\log m)$。

逆元

定义 (逆元)

对正整数 a，a 在模 m 意义下的逆元 a^{-1} 满足 $aa^{-1} \equiv 1 \pmod m$。

求一个数的逆元，可以转化成二元一次不定方程 $ax + my = 1$，然后用之前我们讲过的方法去解即可。时间复杂度 $O(\log m)$。
当 m 是质数的时候，由费马小定理，$a^{m-1} \equiv 1 \pmod m$，所以 a 和 a^{m-2} 互为逆元。可以用快速幂得到。时间复杂度 $O(\log m)$。

逆元

例 (多个数的逆元)

给出正整数 $a_1, \cdots, a_n$，求这些数在模质数 p 意义下的逆元。要求复杂度 $O(n)$。

逆元

例 (多个数的逆元)

给出正整数 $a_1, \cdots, a_n$，求这些数在模质数 p 意义下的逆元。要求复杂度 $O(n)$。

设前缀积 $prod_i = \prod_{j=1}^{i} a_i$。可以 $O(n)$ 递推求出 $1 \sim n$ 每个位置的前缀积。
设前缀积的逆 $iprod_i = prod_i^{-1} = prod_n^{-1} \prod_{j=i+1}^{n} a_i$。在求出 $prod_n$ 后，花 $O(\log p)$ 的时间求出 $iprod_n$。然后可以 $O(n)$ 递推求出 $1 \sim n$ 每个位置的前缀积的逆。
a_i 的逆 $ia_i = a_i^{-1} = iprod_i prod_{i-1}$。所以在求出每个位置的前缀积和前缀积的逆后，可以 $O(n)$ 求出 $1 \sim n$ 每个位置的逆元。总复杂度 $O(n + \log p)$。

线性同余方程组

定义 (线性同余方程组)

有 n 个方程 $x \equiv a_i \pmod{p_i}$，p_i 两两互质，求 x。

定理 (中国剩余定理，CRT)

设 $P = \prod_{i=1}^{n} p_i, w_i = \frac{P}{p_i}$，则方程组的解为 $x \equiv \sum_{i=1}^{n} a_i w_i inv(w_i, p_i) \pmod{P}$，其中，$inv(w_i, p_i)$ 表示 w_i 在模 p_i 意义下的逆。

不难验证这确实是方程组的一个解。直接按照公式计算，复杂度 $O(n \log p)$。

线性同余方程组

定理 (中国剩余定理，CRT)

设 $P=\prod_{i=1}^{n} p_i, w_i=\frac{P}{p_i}$，则方程组的解为
$x\equiv\sum_{i=1}^{n} a_i w_i inv(w_i,p_i) \pmod P$，其中，$inv(w_i,p_i)$ 表示 w_i 在模 p_i 意义下的逆。
并且，CRT 给出的是模 P 意义下的唯一解。

证明.

设 x_1 和 x_2 都是线性同余方程组的解。则对所有的 i，有
$x_1\equiv x_2 \pmod{p_i}$，即 $p_i|(x_1-x_2)$，所以
$\operatorname{lcm}(p_1,\cdots,p_n)|(x_1-x_2)$，又因为 p_i 两两互质，所以
$\operatorname{lcm}(p_1,\cdots,p_n)=P$。 □

线性同余方程组

例 (exCRT)

考虑求方程组 $x \equiv a_1 \pmod{p_1}$ 和 $x \equiv a_2 \pmod{p_2}$ 的解（p_1 和 p_2 不互质）。

设 $d = \gcd(p_1, p_2)$，那么必须 $a_1 \equiv a_2 \pmod{d}$。然后方程组的解一定可以表示为 $wd + (a_1 \bmod d)$。
所以有 $wd + (a_1 \bmod d) \equiv a_1 \pmod{p_1}$，移项有
$wd \equiv a_1 - (a_1 \bmod d) \pmod{p_1}$，即 $w \equiv \frac{a_1}{d} \pmod{\frac{p_1}{d}}$。
类似地，有 $w \equiv \frac{a_2}{d} \pmod{\frac{p_2}{d}}$。这两个同余方程的模数就互质了。
可以用 CRT 求出 w，进而得到原方程组的解。用之前的证明思路，同样可以证明这个方程的解是唯一的。
注意到这样做每次会把两个方程合并成一个模数是它们 lcm 的方程。如果有多个这样的方程可以两两合并。

线性同余方程组

定理 (扩展欧拉定理)

若 $b \geq \varphi(m)$，则 $a^b \equiv a^{b \bmod \varphi(m)+\varphi(m)} \pmod{m}$。

注意到“扩展”之处是不要求 a 和 m 互质。

证明.

设 m 的质因子分解为 $m = \prod p_i^{a_i}$，由之前讲的欧拉函数的性质，$\varphi(p_i^{a_i}) | \varphi(m)$。
考虑 CRT，如果左右在模所有的 $p_i^{a_i}$ 都相等，则它们相等。
当 $p_i | a$ 时，由于 $b \geq \varphi(m) \geq \varphi(p_i^{a_i}) \geq a_i$，于是左右两边都为 0。
否则说明 $\gcd(p_i, a) = 1$，可以用欧拉定理，两边显然是相等的。 □

高次同余初步

例 (bsgs)

设 $\gcd(a, m) = 1$，考虑求方程 $a^x \equiv b \pmod m$ 的解。

令 $B = \lceil \sqrt{m} \rceil$，把 $(ba^0, 0), (ba^1, 1), \cdots, (ba^{B-1}, B-1)$ 全部插入 hash 表里面。
然后令 $d = a^B$，计算 $d^1, d^2, \cdots, d^B$，然后查询 hash 表里面有没有 d^i。如果有，设对应的指数是 k，那就说明 $a^{iB} \equiv ba^k \pmod m$，由于 a, m 互质，所以有 $a^{iB-k} \equiv b \pmod m$。那么就解出了方程的一个解 $x = iB - k$。如果需要所有的解，这里还要继续枚举下去。如果枚举完了都没有，那就无解了。
时间复杂度 $O(\sqrt{m})$。

高次同余初步

例 (exbsgs)

当 $\gcd(a,m) \neq 1$ 时，考虑求方程 $a^x \equiv b \pmod{m}$ 的解。

我们可以考虑从 x 个 a 中拿出 c 个 a，与 b 和 m 消去公因子，直到 a 和 m' 互质为止。
一旦互质了，方程就是 $va^{x-c} = b' \pmod{m'}$。v 是拿出 c 个 a 消去公因子后剩下的东西，b', m' 是消去公因子的 b, m。这时还要求 v 关于 m' 的逆元，方程变为 $a^{x-c} = b' inv(v, m') \pmod{m'}$。此时就可以用 bsgs 做了，答案为 bsgs 的答案加 c。
时间复杂度 $O(\sqrt{m})$。
在实现的时候，可以不求 v 的逆元。bsgs 里面新增一个参数 v，然后哈希表比的时候一边是 va^{iB}，一边是 ba^j。
注意，有可能在消的过程中方程两边就已经相等了，所以必须消去一次就特判一次方程两边（就是 v 和 b'）是不是相等了。如果相等，直接返回此时的 c。

高次同余初步

根据上述算法流程，不难写出代码：

exbsgs

```
ll BSGS(ll a,ll b,ll p,ll v=1){
    ll m=ceil(sqrt(p)),val=1,d;
    for(int i=0;i<m;++i){
        ht.insert(b*val%p,i);
        val=val*a%p;
    }
    d=v*val%p;
    for(int i=1;i<=m;++i){
        int res=ht.query(d);
        if(res!=-1)return i*m-res;
        d=d*val%p;
    }
    return -1;
}
```

高次同余初步

根据上述算法流程，不难写出代码：

exbsgs

```
ll EXBSGS(ll a,ll b,ll p){
    ll t,c=0,v=1;
    while((t=gcd(a,p))!=1){
        if(b%t)return -1;
        p/=t;b/=t;
        v=v*a/t%p; // 注意这个地方p先除了再对v取模
        ++c;if(b==v)return c;
    }
    ll ret=BSGS(a,b,p,v);
    return ret!=-1?ret+c:ret;
}
```

线性同余方程

例 (accoders2241)

`http://www.accoders.com/problem.php?id=2241`

线性同余方程

例 (accoders2241)

`http://www.accoders.com/problem.php?id=2241`

设跳了 t 步，不难写出方程 $mt + x \equiv nt + y \pmod{L}$，然后移项，解就可以了。

线性同余方程组

例 (accoders2244)

`http://www.accoders.com/problem.php?id=2244`

crt 的模板题。

扩展欧拉定理

例 (accoders3704)

给出 p，求 $2^{2^{2^{2^{\cdots}}}} \bmod p$。
数据范围：$p \le 10^7$。

扩展欧拉定理

例 (accoders3704)

给出 p，求 $2^{2^{2^{2^{\cdots}}}} \bmod p$。
数据范围：$p \leq 10^7$。

扩展欧拉定理的一个应用是求无穷层的幂，如这个题。

记 $f(p) = 2^{2^{2^{2^{\cdots}}}} \bmod p$，则 $f(p) \equiv 2^{f(\varphi(p))+\varphi(p)} \pmod p$。
于是我们不断递归至 $\varphi(p) = 1$，此时 $f(\varphi(p)) = 0$。

bsgs

例 (accoders10371 & 9100)

http://www.accoders.com/problem.php?id=10371
http://www.accoders.com/problem.php?id=9100

分别是 bsgs 和 exbsgs 的模板题。

bsgs

例 (accoders10376)

http://www.accoders.com/problem.php?id=10376

这说明了写不带求逆的 bsgs 的好处。

exlucas

定理 (exlucas)

求 $\binom{n}{m} \bmod q$，q 为合数。

首先，我们把 q 分解质因数，那么问题就转化成求 $\binom{n}{m} \bmod p^a$，对每个质数的幂最后再用 CRT 合并。
由于阶乘中可能有质因子 p，所以不能直接求逆。而是应该求下面两个问题：

- $n!$ 中 p 的次数为多少？
- $n!$ 中把所有 p 的质因子提出来之后，$\frac{n!}{p^l} \bmod p^a$ 的值？

exlucas

对于第一个问题，$n!$ 中 p 的次数等于 $n!$ 中含有因子 p 的数的个数加 $n!$ 中含有因子 p^2 的数的个数，以此类推，一直加下去。比较好算。
对于第二个问题，首先把 $n!$ 中 p 的倍数提出来，一共有 $\frac{n}{p}$ 个，都提一个 p 以后，就是一个规模为 $\frac{n}{p}$ 的子问题。由于 p 是质数，所以 $n!$ 中非 p 的倍数就和 p 互质。在模 p^a 意义下会循环 $\frac{n}{p^a}$ 次，还有一个长为 $n \bmod p^a$ 的不完整的循环，这个不完整的循环暴力算。循环节内部的元素都是小于 p^a 且和 p^a 互质的元素，这相当于求模 p^a 意义下缩系元素之积。有结论（hdu4910），当 $p > 2$ 或 $p = 2, a \leq 2$ 时，答案就是 $p^a - 1$，当 $p = 2, a \geq 3$ 时，答案就是 1。当然你也可以暴力算。

Thanks!