数论

ydc

May 19, 2015

这篇课件简单地介绍了 OI 中常考到的一些数论知识

并附上一些题目

目的是让大家对基础的数论算法有个了解

运算器

bzoj 3283

题意：求 $\binom{n}{m} \bmod P$ 与最小的 n 使得 $a^n \bmod P = b$。

说白了就是要介绍两个算法，扩展大步小步和组合数取模。

扩展大步小步

默认大家都会大步小步算法

原理：消因子

$a \bmod b = c$，$d|gcd(a,b,c)$

则有 $(a/d) \bmod (b/d) = c/d$

考虑 $a^n \bmod P = x$

若 a,P 互质，可以直接做，否则，我们取 $d = gcd(a,P)$，若 d 不能被 x 整除则无解，否则等式两边同时除以 d，为 $a/d * a^{n-1} \bmod (P/d) = x/d$。

假设 r 次之后 $(a,P) = 1$，则把 r 轮后的方程的答案后加上 r 即可。r 是 $O(\log P)$ 级别的。另外要注意的是，有可能出现答案小于 r 情况，故要特判。

组合数取模

先总结一下吧

若 n, m 不大，P 是质数，可以用逆元，$O(n) - O(1)$

若 n, m 很大，P 是小质数，可以用 Lucas 定理，$O(P) - O(\log n)$

若 n, m 不大，P 不是质数，可以考虑计算每个质数在 $\binom{n}{m}$ 的指数，复杂度 $O(n)$

若 n, m 很大，P 是质数的若干次方，且不大，可以用接下来讲的算法。

若 n, m, P 都很大，P 是质数，可以看 picks 的毒瘤算法

对于模意义下的问题，如果模数是合数，一般都会分解质因数，考虑模 p^k 的答案，如果要求方案数就是每个的乘积，如果要求一组解就中国剩余定理并起来

将中国剩余定理丢进垃圾桶

说起中国剩余定理，忍不住想插下嘴，介绍一下若干个模数不互质的方程如何并。

考虑如何合并 $x \bmod a_1 = b_1$ 以及 $x \bmod a_2 = b_2$

用扩展欧几里得解如下方程的一组解：
$x \times a_1 + b_1 = y \times a_2 + b_2$，假设解出来一个 x，令
$b = x \times a_1 + b_1$，$a = lcm(a_1, a_2)$，则将方程合并为了
$x \bmod a = b$

如此即使模数不互质也能合并了

组合数取模

计算 $\binom{n}{m} \bmod P$，$P = p^k$，p 是质数

我们考虑把 $n!$ 计算成 $p^n \times s$ 的形式，如此一来就可以计算除法了（这种技巧一般适用于没有加减号的问题）

设 n 是 P 的倍数，不然的话，将零散部分暴力。

在所有 $[kP+1, kP+P-1]$ 这一段里，那些不能被 p 整除的部分的乘积都是相同的，可以算出一个然后快速幂一下。

而所有能被 p 整除的数的乘积，先除去 p，就变成了阶乘的形式，转化为了子问题。

复杂度就是 $O(P)$。

数论之神

bzoj 2219

给定 a, b, P，求 $[0, P-1]$ 里有多少 x 满足 $x^A \bmod P = B$，保证 P 是奇数，多组数据，数字大小是不超过 10^9。

题解

还是分解质因数，那么现在 P 就是某个奇质数的若干次方了。

同时 B 也要模一下当前的模数。令 $P = p^s$

分情况讨论。

情况一

$B=0$。这种情况是很简单的，令 $c=\lceil \frac{s}{A} \rceil$，那么答案就是 p^{s-c}

情况二

$B \neq 0$，$(B,p) = 1$。那么我们求出一个原根 g，就能用 g^x 表示 B 了。

接着相当于求一个不定方程的解数，我们知道模 P 意义下 $ax = b$ 这个方程如果有解，解数是 $gcd(a,P)$，用这个性质即可。

情况三

设 $B \neq 0, gcd(B,p) \neq 1$

设 $B = p^{s_1} * u$。

根据消因子，可以写成 $x^A/p^{s_1} \equiv u(\mathrm{mod} p^{s-s_1})$。

可以发现，如果 A 不能被 s_1 整除，是无解的。设 $A = s_1/t$，则写成 $(x/p^t)^A \equiv u(\mathrm{mod} p^{s-s_1})$，设 $X = x/p^t$，变成了 $X \equiv u(\mathrm{mod} p^{s-s_1})$。

新方程与原方程的解一一对应，而 $gcd(u,p) = 1$，转化为了情况二

小总结

这类模意义下的求解数问题，通常都是先分解质因数，对于每个 p^k 进行讨论。（此外还有 dzy loves math3 等题目）

此外，本题还用到了消因子、原根等数论技巧，可以说这题囊括了很多的数论知识。

再看扩展欧几里得

欧几里得算法是非常基础的联赛内容。它的时间复杂度基于 (a, b) 变为 $(a \bmod b, b)$ 这种变换只会有 $O(\log a)$ 次。

事实上还有很多算法，也是基于这种变换次数很少而提出。

再看扩展欧几里得

求 $\sum_{x=1}^{n} \lfloor \frac{Ax+B}{C} \rfloor$

再看扩展欧几里得

求 $\sum_{x=1}^{n} \lfloor \frac{Ax+B}{C} \rfloor$

求最小的 x，满足 $l \leq ax \bmod p \leq r$

再看扩展欧几里得

求 $\sum_{x=1}^{n} \lfloor \frac{Ax+B}{C} \rfloor$

求最小的 x，满足 $l \le ax \bmod p \le r$

求模合数意义下的行列式

wc 2014 时空穿梭

一个 n 维空间，第 i 维长度是 m_i，要你对于每两个整点 A, B，求他们连线段上的整点数。

题解

一个二维向量 (x, y) 的整点数是 $gcd(x, y) + 1$

很容易扩展为 n 维向量 $(a_1, a_2, \cdots, a_n)$ 的整点数是 $gcd(a_1, a_2, \cdots, a_n) + 1$

先看二维情形，答案是:

$$\sum_{x=1}^{m_1} \sum_{y=1}^{m_2} (m_1 - x) \times (m_2 - y) \times \binom{gcd(x,y)-1}{c-2}$$

来对其化简。

题解

$$
\begin{aligned}
ans &= \sum_{x=1}^{m_1}\sum_{y=1}^{m_2}(m_1-x)\times(m_2-y)\times\binom{gcd(x,y)-1}{c-2} \\
&= \sum_{d=1}^{m_1}\binom{d-1}{c-2}\sum_{i=1}^{\lfloor\frac{m_1}{d}\rfloor}\sum_{j=1}^{\lfloor\frac{m_2}{d}\rfloor}(m_1-id)(m_2-jd)e(gcd(i,j)) \\
&= \sum_{d=1}^{m_1}\binom{d-1}{c-2}\sum_{i=1}^{\lfloor\frac{m_1}{d}\rfloor}\sum_{j=1}^{\lfloor\frac{m_2}{d}\rfloor}\sum_{d'|gcd(i,j)}\mu(d')(m_1-id)(m_2-jd) \\
&= \sum_{d=1}^{m_1}\binom{d-1}{c-2}\sum_{d'}^{m_1}\sum_{i=1}^{\lfloor\frac{m_1}{dd'}\rfloor}(m_1-idd')\sum_{j=1}^{\lfloor\frac{m_2}{dd'}\rfloor}(m_2-jdd')
\end{aligned}
$$

题解

积性函数题有个技巧，一旦遇到了枚举 d，枚举 d'，并涉及到了 dd' 时，一定要改变思路，变成枚举 $D = dd'$，如此一来，就能得到卷积形式。

我们很容易将二位情况扩展为 n 维，那么就能有如下变形：

$$
\begin{aligned}
ans &= \sum_{D=1}^{m_1} \prod_{i=1}^{n} \sum_{j=1}^{\lfloor \frac{m_i}{D} \rfloor} (m_i - jD) \sum_{d|D} \binom{d-1}{c-2} \mu(\frac{D}{d}) \\
&= \sum_{D=1}^{m_1} \prod_{i=1}^{n} \frac{(2m_i - \lfloor \frac{m_i}{D} + 1 \rfloor D) \lfloor \frac{m_i}{D} \rfloor}{2} \sum_{d|D} \binom{d-1}{c-2} \mu(\frac{D}{d})
\end{aligned}
$$

到此为止就和我们常见的积性函数题很像了，只要后面的 $\sum_{d|D}\binom{d-1}{c-2}\mu(\frac{D}{d})$ 是积性函数，问题就迎刃而解。

而事实上组合数并不是积性函数，我们需要做点处理。我们知道组合数 $\binom{n}{m}$ 是一个 m 次多项式，所以拆成 m 个单项式的和，就变成 m 个积性函数的和了。

套用经典算法，枚举 $2n\sqrt{m}$ 段，$O(n^2)$ 的展开连乘式，最后单次询问的复杂度是 $O(\sqrt{m}n^3)$

小总结

这题是个不错的积性函数题，基本上积性函数题推导的思想都用上了

遇到枚举了 d, d' 要用到 dd' 时，我们要反过来，先枚举 $D = dd'$，再枚举 $d|D$，原因是积性函数的卷积还是积性函数

如果遇到了像组合数啊，$\sum_{i=1}^{n} i^k$ 这种不是积性函数的东西，要先想想能不能表示成若干个积性函数的和

理性愉悦

求 $\sum_{i=1}^{n}\sum_{j=1}^{m} gcd(i,j)$

$n, m \le 10^{10}$

理性愉悦

$$\Phi(n) = \sum_{i=1}^{n} \phi(n)$$

理性愉悦

$$\Phi(n) = \sum_{i=1}^{n} \phi(n)$$

$$g(n) = \sum_{d|n} \phi(d)$$

理性愉悦

$$\Phi(n) = \sum_{i=1}^{n} \phi(n)$$

$$g(n) = \sum_{d|n} \phi(d)$$

$$G(n) = \sum_{i=1}^{n} \sum_{d|i} \phi(d) = \sum_{i=1}^{n} \sum_{j=1}^{\lfloor \frac{n}{i} \rfloor} \phi(j)$$

理性愉悦

$$\Phi(n) = \sum_{i=1}^{n} \phi(n)$$

$$g(n) = \sum_{d|n} \phi(d)$$

$$G(n) = \sum_{i=1}^{n} \sum_{d|i} \phi(d) = \sum_{i=1}^{n} \sum_{j=1}^{\lfloor \frac{n}{i} \rfloor} \phi(j)$$

$$G(n) = \sum_{i=1}^{n} \Phi(\tfrac{n}{i})$$

理性愉悦

$$\Phi(n)=\sum_{i=1}^{n}\phi(n)$$

$$g(n)=\sum_{d|n}\phi(d)$$

$$G(n)=\sum_{i=1}^{n}\sum_{d|i}\phi(d)=\sum_{i=1}^{n}\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}\phi(j)$$

$$G(n)=\sum_{i=1}^{n}\Phi(\tfrac{n}{i})$$

$$\Phi(n)=\sum_{i=1}^{n}\Phi(\lfloor\tfrac{n}{i}\rfloor)-\sum_{i=2}^{n}\Phi(\lfloor\tfrac{n}{i}\rfloor)=G(n)-\sum_{i=2}^{n}\Phi(\lfloor\tfrac{n}{i}\rfloor)$$

理性愉悦

$$\Phi(n) = \sum_{i=1}^{n} \phi(n)$$

$$g(n) = \sum_{d|n} \phi(d)$$

$$G(n) = \sum_{i=1}^{n} \sum_{d|i} \phi(d) = \sum_{i=1}^{n} \sum_{j=1}^{\lfloor \frac{n}{i} \rfloor} \phi(j)$$

$$G(n) = \sum_{i=1}^{n} \Phi(\tfrac{n}{i})$$

$$\Phi(n) = \sum_{i=1}^{n} \Phi(\lfloor \tfrac{n}{i} \rfloor) - \sum_{i=2}^{n} \Phi(\lfloor \tfrac{n}{i} \rfloor) = G(n) - \sum_{i=2}^{n} \Phi(\lfloor \tfrac{n}{i} \rfloor)$$

理性愉悦

$$g(n)=\sum_{d|n}\phi(d)=n, G(n)=\frac{n(n+1)}{2}$$

理性愉悦

$$g(n) = \sum_{d|n} \phi(d) = n, G(n) = \frac{n(n+1)}{2}$$

$$\Phi(n) = \frac{n(n+1)}{2} - \sum_{i=2}^{n} \Phi(\lfloor \frac{n}{i} \rfloor)$$

理性愉悦

可以先筛出前 $n^{2/3}$ 的前缀和函数

然后根据这个式子记忆化下去

复杂度：$O(n^{\frac{2}{3}})$

小总结

令原函数为 $f(x)$，前缀和函数为 $F(x)$

令 $g(n) = \sum_{d|n} f(d), G(x) = \sum_{i=1}^{n} g(i)$

则 $F(n) = G(n) - \sum_{i=2}^{n} F(\lfloor \frac{n}{i} \rfloor)$

若原函数是 ϕ，则 $G(n) = \frac{n(n+1)}{2}$

若原函数是 μ，则 $G(n) = 1$

能这么做的题还有，SDOI2015 第一轮 day2 T2，crash 的数字表格，时间复杂度都是 $O(n^{2/3})$

有的题目由于不能方便的算 G，所以复杂度是 $O(n^{3/4})$