

多项式及求和

杜瑜皓

浙江省镇海中学

January 26, 2013

首先我们来看一个问题

$$\sum_{i=0}^n i^d \bmod m$$

首先我们来看一个问题

$$\sum_{i=0}^n i^d \bmod m$$

- $d \leq 50, n \leq 10^{18}, m \leq 10^{18}$

首先我们来看一个问题

$$\sum_{i=0}^n i^d \bmod m$$

- $d \leq 50, n \leq 10^{18}, m \leq 10^{18}$
- $d \leq 200, n \leq 10^{10000}, m \leq 10^{18}, m$ 为素数

首先我们来看一个问题

$$\sum_{i=0}^n i^d \bmod m$$

- $d \leq 50, n \leq 10^{18}, m \leq 10^{18}$
- $d \leq 200, n \leq 10^{10000}, m \leq 10^{18}, m$ 为素数
- $d \leq 2000, n \leq 10^{10000}, m \leq 10^{18}, m$ 为素数

首先我们来看一个问题

$$\sum_{i=0}^n i^d \bmod m$$

- $d \leq 50, n \leq 10^{18}, m \leq 10^{18}$
- $d \leq 200, n \leq 10^{10000}, m \leq 10^{18}, m$ 为素数
- $d \leq 2000, n \leq 10^{10000}, m \leq 10^{18}, m$ 为素数
- $d \leq 200000, n \leq 10^{10000}, m \leq 10^{18}, m$ 为素数

$$1. d \leq 50, n \leq 10^{18}, m \leq 10^{18}$$

1. $d \leq 50, n \leq 10^{18}, m \leq 10^{18}$

- 我们令

$$S_d(n) = \sum_{i=0}^{n-1} i^d$$

1. $d \leq 50, n \leq 10^{18}, m \leq 10^{18}$

- 我们令

$$S_d(n) = \sum_{i=0}^{n-1} i^d$$

- 构造向量 $V_n = \{S_d(n), n^0, n^1, \dots, n^d\}^T$

1. $d \leq 50, n \leq 10^{18}, m \leq 10^{18}$

- 我们令

$$S_d(n) = \sum_{i=0}^{n-1} i^d$$

- 构造向量 $V_n = \{S_d(n), n^0, n^1, \dots, n^d\}^T$
- 转移：

$$S_d(n+1) = S_d(n) + n^d$$

1. $d \leq 50, n \leq 10^{18}, m \leq 10^{18}$

- 我们令

$$S_d(n) = \sum_{i=0}^{n-1} i^d$$

- 构造向量 $V_n = \{S_d(n), n^0, n^1, \dots, n^d\}^T$
- 转移：

$$S_d(n+1) = S_d(n) + n^d$$

-

$$(n+1)^i = \sum_{j=0}^i \binom{i}{j} n^j$$

1. $d \leq 50, n \leq 10^{18}, m \leq 10^{18}$

- 我们令

$$S_d(n) = \sum_{i=0}^{n-1} i^d$$

- 构造向量 $V_n = \{S_d(n), n^0, n^1, \dots, n^d\}^T$
- 转移：

$$S_d(n+1) = S_d(n) + n^d$$

-

$$(n+1)^i = \sum_{j=0}^i \binom{i}{j} n^j$$

-

$$V_{n+1} = V_n * A$$

- A 是一个 $(d+2) \times (d+2)$ 的矩阵，具体系数可以通过前面的转移方程得出。

- A 是一个 $(d+2) * (d+2)$ 的矩阵，具体系数可以通过前面的转移方程得出。
- 用矩阵乘法加快速幂优化，那么就可以在 $O(d^3 \log n)$ 的时间内解决。

- A 是一个 $(d+2) * (d+2)$ 的矩阵，具体系数可以通过前面的转移方程得出。
- 用矩阵乘法加快速幂优化，那么就可以在 $O(d^3 \log n)$ 的时间内解决。
- 这里并没有除法运算，所以和 m 取值的关系并不大。

2. $d \leq 200, n \leq 10^{10000}, m \leq 10^{18}$, m 为素数

2. $d \leq 200, n \leq 10^{10000}, m \leq 10^{18}$, m 为素数

- 我们可以证明 $S_d(n)$ 是关于 n 的 $d+1$ 次的多项式。

2. $d \leq 200, n \leq 10^{10000}, m \leq 10^{18}$, m 为素数

- 我们可以证明 $S_d(n)$ 是关于 n 的 $d+1$ 次的多项式。
- 进行一下简单的数学推导

2. $d \leq 200, n \leq 10^{10000}, m \leq 10^{18}$, m 为素数

- 我们可以证明 $S_d(n)$ 是关于 n 的 $d+1$ 次的多项式。
- 进行一下简单的数学推导
- $d=0$ 时显然成立

2. $d \leq 200, n \leq 10^{10000}, m \leq 10^{18}$, m 为素数

- 我们可以证明 $S_d(n)$ 是关于 n 的 $d+1$ 次的多项式。
- 进行一下简单的数学推导
- $d=0$ 时显然成立
- 假设 $d=k$ 时成立

2. $d \leq 200, n \leq 10^{10000}, m \leq 10^{18}$, m 为素数

- 我们可以证明 $S_d(n)$ 是关于 n 的 $d+1$ 次的多项式。
- 进行一下简单的数学推导
- $d=0$ 时显然成立
- 假设 $d=k$ 时成立
- 当 $d=k+1$ 时

$$(j+1)^{d+1} - j^{d+1} = \sum_{i=0}^d \binom{d+1}{i} j^i$$



$$(j+1)^{d+1} - j^{d+1} = \sum_{i=0}^d \binom{d+1}{i} j^i$$



$$(j+1)^{d+1} - j^{d+1} = \sum_{i=0}^d \binom{d+1}{i} j^i$$

- 把 j 从0到 $n-1$ 求和

$$\sum_{j=0}^{n-1} (j+1)^{d+1} - j^{d+1} = \sum_{j=0}^{n-1} \sum_{i=0}^d \binom{d+1}{i} j^i$$



$$(j+1)^{d+1} - j^{d+1} = \sum_{i=0}^d \binom{d+1}{i} j^i$$

- 把 j 从0到 $n-1$ 求和

$$\sum_{j=0}^{n-1} (j+1)^{d+1} - j^{d+1} = \sum_{j=0}^{n-1} \sum_{i=0}^d \binom{d+1}{i} j^i$$

- 也就是

$$n^{d+1} = \sum_{i=0}^d \binom{d+1}{i} S_i(n)$$



$$(j+1)^{d+1} - j^{d+1} = \sum_{i=0}^d \binom{d+1}{i} j^i$$

- 把 j 从0到 $n-1$ 求和

$$\sum_{j=0}^{n-1} (j+1)^{d+1} - j^{d+1} = \sum_{j=0}^{n-1} \sum_{i=0}^d \binom{d+1}{i} j^i$$

- 也就是

$$n^{d+1} = \sum_{i=0}^d \binom{d+1}{i} S_i(n)$$

- 即

$$(d+1) * S_d(n) = n^{d+1} - \sum_{i=0}^{d-1} \binom{d+1}{i} S_i(n)$$

- $$(d+1) * S_d(n) = n^{d+1} - \sum_{i=0}^{d-1} \binom{d+1}{j} S_i(n)$$



$$(d+1) * S_d(n) = n^{d+1} - \sum_{i=0}^{d-1} \binom{d+1}{j} S_i(n)$$

- 右边显然是一个次数不超过 $d+1$ 次的多项式，并且通过这个式子我们可以递推算出 $S_d(n)$ 。



$$(d+1) * S_d(n) = n^{d+1} - \sum_{i=0}^{d-1} \binom{d+1}{j} S_i(n)$$

- 右边显然是一个次数不超过 $d+1$ 次的多项式，并且通过这个式子我们可以递推算出 $S_d(n)$ 。
- 时间复杂度为 $O(d^3 + \log n)$ 。

3. $d \leq 2000, n \leq 10^{10000}, m \leq 10^{18}$, m 为素数

- 我们定义伯努利数

$$\frac{x}{e^x - 1} = \sum_{n=0}^{\infty} \frac{B_n}{n!} x^n$$

3. $d \leq 2000, n \leq 10^{10000}, m \leq 10^{18}$, m 为素数

- 我们定义伯努利数

$$\frac{x}{e^x - 1} = \sum_{n=0}^{\infty} \frac{B_n}{n!} x^n$$

- 定义伯努利多项式

$$\sum_{n=0}^{\infty} \frac{\beta_n(t)}{n!} x^n = \frac{x}{e^x - 1} e^{tx} = \left(\sum_{n=0}^{\infty} \frac{B_n}{n!} x^n \right) \left(\sum_{n=0}^{\infty} \frac{t^n}{n!} x^n \right)$$

3. $d \leq 2000, n \leq 10^{10000}, m \leq 10^{18}$, m 为素数

- 我们定义伯努利数

$$\frac{x}{e^x - 1} = \sum_{n=0}^{\infty} \frac{B_n}{n!} x^n$$

- 定义伯努利多项式

$$\sum_{n=0}^{\infty} \frac{\beta_n(t)}{n!} x^n = \frac{x}{e^x - 1} e^{tx} = \left(\sum_{n=0}^{\infty} \frac{B_n}{n!} x^n \right) \left(\sum_{n=0}^{\infty} \frac{t^n}{n!} x^n \right)$$

- 也就是

$$\beta_n(t) = \sum_{k=0}^n \binom{n}{k} B_{n-k} t^k$$

$$\begin{aligned}
 \sum_{n=0}^{\infty} \frac{\beta_n(t+1) - \beta_n(t)}{n!} x^n &= \frac{x e^{(t+1)x}}{e^x - 1} - \frac{x e^{tx}}{e^x - 1} \\
 &= \frac{x e^{tx} (e^x - 1)}{e^x - 1} = x e^{tx} = x \sum_{n=0}^{\infty} \frac{t^n}{n!} x^n
 \end{aligned}$$

$$\begin{aligned}
 \sum_{n=0}^{\infty} \frac{\beta_n(t+1) - \beta_n(t)}{n!} x^n &= \frac{xe^{(t+1)x}}{e^x - 1} - \frac{xe^{tx}}{e^x - 1} \\
 &= \frac{xe^{tx}(e^x - 1)}{e^x - 1} = xe^{tx} = x \sum_{n=0}^{\infty} \frac{t^n}{n!} x^n
 \end{aligned}$$

- 观察两边 x^{n+1} 的系数，即得

$$\frac{\beta_{n+1}(t+1) - \beta_{n+1}(t)}{(n+1)!} = \frac{t^n}{n!}$$

$$\begin{aligned} \sum_{n=0}^{\infty} \frac{\beta_n(t+1) - \beta_n(t)}{n!} x^n &= \frac{x e^{(t+1)x}}{e^x - 1} - \frac{x e^{tx}}{e^x - 1} \\ &= \frac{x e^{tx} (e^x - 1)}{e^x - 1} = x e^{tx} = x \sum_{n=0}^{\infty} \frac{t^n}{n!} x^n \end{aligned}$$

- 观察两边 x^{n+1} 的系数, 即得

$$\frac{\beta_{n+1}(t+1) - \beta_{n+1}(t)}{(n+1)!} = \frac{t^n}{n!}$$

- 就是

$$\beta_{n+1}(t+1) - \beta_{n+1}(t) = (n+1)t^n$$

$$\begin{aligned}\sum_{n=0}^{\infty} \frac{\beta_n(t+1) - \beta_n(t)}{n!} x^n &= \frac{xe^{(t+1)x}}{e^x - 1} - \frac{xe^{tx}}{e^x - 1} \\ &= \frac{xe^{tx}(e^x - 1)}{e^x - 1} = xe^{tx} = x \sum_{n=0}^{\infty} \frac{t^n}{n!} x^n\end{aligned}$$

- 观察两边 x^{n+1} 的系数, 即得

$$\frac{\beta_{n+1}(t+1) - \beta_{n+1}(t)}{(n+1)!} = \frac{t^n}{n!}$$

- 就是

$$\beta_{n+1}(t+1) - \beta_{n+1}(t) = (n+1)t^n$$

- 所以

$$S_d(n) = \frac{\beta_{d+1}(n) - \beta_{d+1}(0)}{d+1}$$

- 经化简可得

$$\sum_{k=0}^{m-1} k^n = \frac{1}{n+1} \sum_{k=0}^n \binom{n+1}{k} B_k m^{n+1-k}$$

- 经化简可得

$$\sum_{k=0}^{m-1} k^n = \frac{1}{n+1} \sum_{k=0}^n \binom{n+1}{k} B_k m^{n+1-k}$$

- 在这个式子中，令 $m=1$ 且 $n \neq 0$ ，那么就有

$$\sum_{k=0}^n \binom{n+1}{k} B_k = 0 \Rightarrow B_n = -\frac{1}{n+1} \sum_{k=0}^{n-1} \binom{n+1}{k} B_k$$

- 经化简可得

$$\sum_{k=0}^{m-1} k^n = \frac{1}{n+1} \sum_{k=0}^n \binom{n+1}{k} B_k m^{n+1-k}$$

- 在这个式子中，令 $m=1$ 且 $n \neq 0$ ，那么就有

$$\sum_{k=0}^n \binom{n+1}{k} B_k = 0 \Rightarrow B_n = -\frac{1}{n+1} \sum_{k=0}^{n-1} \binom{n+1}{k} B_k$$

- 另外证明可以参见顾宇宙大神JZPKIL的题解或《具体数学》

- 于是我们可以在 $O(d^2)$ 的时间内算出伯努利数，然后算出 $S_d(n)$ 时间复杂度为 $O(d^2 + \log n)$ 。

4. $d \leq 200000, n \leq 10^{10000}, m \leq 10^{18}$, m 为素数

4. $d \leq 200000$, $n \leq 10^{10000}$, $m \leq 10^{18}$, m 为素数

- 关注 $S_d(x)$ 这个多项式本身, 我们可以在 $O(d \log d)$ 的时间内算出 $S_d(1), S_d(2), \dots, S_d(d+1)$

4. $d \leq 200000$, $n \leq 10^{10000}$, $m \leq 10^{18}$, m 为素数

- 关注 $S_d(x)$ 这个多项式本身，我们可以在 $O(d \log d)$ 的时间内算出 $S_d(1), S_d(2), \dots, S_d(d+1)$
- 而确定了 $S_d(1), S_d(2), \dots, S_d(d+1)$ 也就代表确定了这个多项式。

4. $d \leq 200000$, $n \leq 10^{10000}$, $m \leq 10^{18}$, m 为素数

- 关注 $S_d(x)$ 这个多项式本身，我们可以在 $O(d \log d)$ 的时间内算出 $S_d(1), S_d(2), \dots, S_d(d+1)$
- 而确定了 $S_d(1), S_d(2), \dots, S_d(d+1)$ 也就代表确定了这个多项式。
- 多项式的系数能在 $O(d^2)$ 的时间内计算。

4. $d \leq 200000$, $n \leq 10^{10000}$, $m \leq 10^{18}$, m 为素数

- 关注 $S_d(x)$ 这个多项式本身，我们可以在 $O(d \log d)$ 的时间内算出 $S_d(1), S_d(2), \dots, S_d(d+1)$
- 而确定了 $S_d(1), S_d(2), \dots, S_d(d+1)$ 也就代表确定了这个多项式。
- 多项式的系数能在 $O(d^2)$ 的时间内计算。
- 我们定义

$$S_{d-1}(x) = P(x) = \sum_{k=0}^d \binom{x}{k} a_k$$


4. $d \leq 200000$, $n \leq 10^{10000}$, $m \leq 10^{18}$, m 为素数

- 关注 $S_d(x)$ 这个多项式本身，我们可以在 $O(d \log d)$ 的时间内算出 $S_d(1), S_d(2), \dots, S_d(d+1)$
- 而确定了 $S_d(1), S_d(2), \dots, S_d(d+1)$ 也就代表确定了这个多项式。
- 多项式的系数能在 $O(d^2)$ 的时间内计算。
- 我们定义

$$S_{d-1}(x) = P(x) = \sum_{k=0}^d \binom{x}{k} a_k$$

- 经过二项式反演可得

$$a_k = \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} P(j)$$


$$a_k = \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} P(j)$$



$$a_k = \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} P(j)$$

• 也就是

$$\frac{a_k}{k!} = \sum_{j=0}^k \frac{(-1)^{k-j}}{(k-j)!} \cdot \frac{P(j)}{j!}$$



$$a_k = \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} P(j)$$

- 也就是

$$\frac{a_k}{k!} = \sum_{j=0}^k \frac{(-1)^{k-j}}{(k-j)!} \cdot \frac{P(j)}{j!}$$

- 注意这是一个卷积的形式，我们可以用FFT在 $O(d \log d)$ 的时间内算出来。



$$a_k = \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} P(j)$$

- 也就是

$$\frac{a_k}{k!} = \sum_{j=0}^k \frac{(-1)^{k-j}}{(k-j)!} \cdot \frac{P(j)}{j!}$$

- 注意这是一个卷积的形式，我们可以用FFT在 $O(d \log d)$ 的时间内算出来。
- FFT?

•

$$a_k = \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} P(j)$$



$$a_k = \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} P(j)$$

• 那么就有

$$\begin{aligned} P(n) &= \sum_{k=0}^d \binom{n}{k} a_k = \sum_{k=0}^d \binom{n}{k} \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} P(j) \\ &= \sum_{j=0}^d P(j) \sum_{k=j}^d (-1)^{k-j} \binom{n}{k} \binom{k}{j} \end{aligned}$$



$$\begin{aligned}\sum_{k=j}^d (-1)^{k-j} \binom{n}{k} \binom{k}{j} &= \sum_{k=j}^d (-1)^{k-j} \frac{n! k!}{k! (n-k)! j! (k-j)!} \\&= \sum_{k=j}^d (-1)^{k-j} \frac{n!}{j! (n-j)!} \frac{(n-j)!}{(n-k)! (k-j)!} = \sum_{k=j}^d (-1)^{k-j} \binom{n}{j} \binom{n-j}{k-j} \\&= \binom{n}{j} \sum_{k=0}^{d-j} (-1)^k \binom{n-j}{k}\end{aligned}$$

$$\begin{aligned}
\sum_{k=j}^d (-1)^{k-j} \binom{n}{k} \binom{k}{j} &= \sum_{k=j}^d (-1)^{k-j} \frac{n! k!}{k! (n-k)! j! (k-j)!} \\
&= \sum_{k=j}^d (-1)^{k-j} \frac{n!}{j! (n-j)!} \frac{(n-j)!}{(n-k)! (k-j)!} = \sum_{k=j}^d (-1)^{k-j} \binom{n}{j} \binom{n-j}{k-j} \\
&= \binom{n}{j} \sum_{k=0}^{d-j} (-1)^k \binom{n-j}{k}
\end{aligned}$$

$$\begin{aligned}
\sum_{i=0}^k (-1)^i \binom{n}{i} &= \binom{n}{0} - \binom{n}{1} + \dots = \binom{n-1}{0} - \binom{n}{1} + \dots \\
&= -\binom{n-1}{1} + \binom{n}{2} + \dots = \binom{n-1}{2} - \binom{n}{3} + \dots = (-1)^k \binom{n-1}{k}
\end{aligned}$$

$$\sum_{k=j}^d (-1)^{k-j} \binom{n}{k} \binom{k}{j} = (-1)^{d-j} \binom{n-j-1}{d-j} \binom{n}{j}$$

$$\sum_{k=j}^d (-1)^{k-j} \binom{n}{k} \binom{k}{j} = (-1)^{d-j} \binom{n-j-1}{d-j} \binom{n}{j}$$

$$\begin{aligned} P(n) &= \sum_{j=0}^d (-1)^{d-j} P(j) \binom{n-j-1}{d-j} \binom{n}{j} \\ &= \sum_{j=0}^d (-1)^{d-j} P(j) \frac{(n-j-1)!n!}{(d-j)!(n-d-1)!(n-j)!j!} \\ &= \sum_{j=0}^d (-1)^{d-j} P(j) \frac{n(n-1)\dots(n-d)}{(d-j)!j!(n-j)} \end{aligned}$$

- 这样我们在知道 $P(0), P(1), \dots, P(d)$ 的情况下，可以在 $O(d)$ 的时间复杂度内算出 $P(n)$

- 这样我们在知道 $P(0), P(1), \dots, P(d)$ 的情况下，可以在 $O(d)$ 的时间复杂度内算出 $P(n)$
- 对于本题， $S_d(n)$ 的计算是瓶颈，注意到 i^d 是积性函数，那么只要算出那些素数的幂次就可以算出所有的 i^d 的值了。

- 这样我们在知道 $P(0), P(1), \dots, P(d)$ 的情况下，可以在 $O(d)$ 的时间复杂度内算出 $P(n)$
- 对于本题， $S_d(n)$ 的计算是瓶颈，注意到 i^d 是积性函数，那么只要算出那些素数的幂次就可以算出所有的 i^d 的值了。
- 由于不超过 d 的素数是 $O(d/\log d)$ 个，而快速幂的复杂度为 $O(\log d)$ ，那么时间复杂度就是 $O(d)$

- 这样我们在知道 $P(0), P(1), \dots, P(d)$ 的情况下，可以在 $O(d)$ 的时间复杂度内算出 $P(n)$
- 对于本题， $S_d(n)$ 的计算是瓶颈，注意到 i^d 是积性函数，那么只要算出那些素数的幂次就可以算出所有的 i^d 的值了。
- 由于不超过 d 的素数是 $O(d/\log d)$ 个，而快速幂的复杂度为 $O(\log d)$ ，那么时间复杂度就是 $O(d)$
- 我们就在 $O(d)$ 的时间内把这个问题解决了。

$$4^*.d \leq 200000, n \leq 10^{10000}, m \leq 10^{18}$$

$$4^*.d \leq 200000, n \leq 10^{10000}, m \leq 10^{18}$$

- m 不是素数，那么对组合数计算会带来困难。

$$4^*.d \leq 200000, n \leq 10^{10000}, m \leq 10^{18}$$

- m 不是素数，那么对组合数计算会带来困难。
- 我们令 $m = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k} M$ ，其中 $p_i \leq d + 1$ 。

$$4^*.d \leq 200000, n \leq 10^{10000}, m \leq 10^{18}$$

- m 不是素数，那么对组合数计算会带来困难。
- 我们令 $m = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k} M$ ，其中 $p_i \leq d + 1$ 。
- 我们可以分别求出 $S_d(n)$ 对 $p_1^{a_1}, p_2^{a_2}, \dots, p_k^{a_k}, M$ 取模，然后用中国剩余定理进行合并即可。

$4^*. d \leq 200000, n \leq 10^{10000}, m \leq 10^{18}$

- m 不是素数，那么对组合数计算会带来困难。
- 我们令 $m = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k} M$ ，其中 $p_i \leq d + 1$ 。
- 我们可以分别求出 $S_d(n)$ 对 $p_1^{a_1}, p_2^{a_2}, \dots, p_k^{a_k}, M$ 取模，然后用中国剩余定理进行合并即可。
- 对 M 取模我们可以根据上述的方法进行处理，因为 M 没有小于 d 的因子，那么 $(d-j)!$ 和 M 一定是互质的，而 $n-j$ 一定是 $n, n-1, \dots, n-d$ 中的某一个，只要约去即可。

- 接下来要求的是 $S_d(n)$ 对 p^a 取模，令 $n - 1 = q'p^a + r'$ ，那么就有 $S_d(n) \equiv q'S_d(p^a) + S_d(r') \pmod{p^a}$ ，所以只要考虑 $n \leq p^a$ 时即可。

- 接下来要求的是 $S_d(n)$ 对 p^a 取模，令 $n-1 = q'p^a + r'$ ，那么就有 $S_d(n) \equiv q'S_d(p^a) + S_d(r') \pmod{p^a}$ ，所以只要考虑 $n \leq p^a$ 时即可。
- 令 $n-1 = qp + r$ ，那么 $S_d(n) = \sum_{i=0}^{qp-1} i^d + \sum_{i=qp}^{n-1} i^d$

- 接下来要求的是 $S_d(n)$ 对 p^a 取模，令 $n-1 = q'p^a + r'$ ，那么就有 $S_d(n) \equiv q'S_d(p^a) + S_d(r') \pmod{p^a}$ ，所以只要考虑 $n \leq p^a$ 时即可。
- 令 $n-1 = qp + r$ ，那么 $S_d(n) = \sum_{i=0}^{qp-1} i^d + \sum_{i=qp}^{n-1} i^d$
- 对于 $\sum_{i=qp}^{n-1} i^d$ 它的个数不超过 p ，也可通过暴力直接计算。

- 接下来要求的是 $S_d(n)$ 对 p^a 取模，令 $n-1 = q'p^a + r'$ ，那么就有 $S_d(n) \equiv q'S_d(p^a) + S_d(r') \pmod{p^a}$ ，所以只要考虑 $n \leq p^a$ 时即可。
- 令 $n-1 = qp + r$ ，那么 $S_d(n) = \sum_{i=0}^{qp-1} i^d + \sum_{i=qp}^{n-1} i^d$
- 对于 $\sum_{i=qp}^{n-1} i^d$ 它的个数不超过 p ，也可通过暴力直接计算。
-

$$\sum_{i=0}^{qp-1} i^d = \sum_{i=0}^{q-1} \sum_{j=0}^{p-1} (ip + j)^d = \sum_{i=0}^{q-1} \sum_{j=0}^{p-1} \sum_{k=0}^d \binom{d}{k} (ip)^k j^{d-k}$$

- 接下来要求的是 $S_d(n)$ 对 p^a 取模，令 $n-1 = q'p^a + r'$ ，那么就有 $S_d(n) \equiv q'S_d(p^a) + S_d(r') \pmod{p^a}$ ，所以只要考虑 $n \leq p^a$ 时即可。
- 令 $n-1 = qp + r$ ，那么 $S_d(n) = \sum_{i=0}^{qp-1} i^d + \sum_{i=qp}^{n-1} i^d$
- 对于 $\sum_{i=qp}^{n-1} i^d$ 它的个数不超过 p ，也可通过暴力直接计算。
-

$$\sum_{i=0}^{qp-1} i^d = \sum_{i=0}^{q-1} \sum_{j=0}^{p-1} (ip + j)^d = \sum_{i=0}^{q-1} \sum_{j=0}^{p-1} \sum_{k=0}^d \binom{d}{k} (ip)^k j^{d-k}$$

- 由于对 p^a 取模，那么 $(ip)^k$ 中如果 $k \geq a$ 就可以忽略。

- 也就是

$$\begin{aligned}\sum_{i=0}^{qp-1} i^d &= \sum_{i=0}^{q-1} \sum_{j=0}^{p-1} \sum_{k=0}^{a-1} \binom{d}{k} (ip)^k j^{d-k} \\ &= \sum_{k=0}^{a-1} \binom{d}{k} p^k \sum_{j=0}^{p-1} j^{d-k} \sum_{i=0}^{q-1} i^k\end{aligned}$$

- 也就是

$$\begin{aligned}\sum_{i=0}^{qp-1} i^d &= \sum_{i=0}^{q-1} \sum_{j=0}^{p-1} \sum_{k=0}^{a-1} \binom{d}{k} (ip)^k j^{d-k} \\ &= \sum_{k=0}^{a-1} \binom{d}{k} p^k \sum_{j=0}^{p-1} j^{d-k} \sum_{i=0}^{q-1} i^k\end{aligned}$$

- $\sum_{j=0}^{p-1} j^{d-k}$, $\sum_{i=0}^{q-1} i^k$ 两个互相独立，可以分开求和。

- 也就是

$$\begin{aligned}\sum_{i=0}^{qp-1} i^d &= \sum_{i=0}^{q-1} \sum_{j=0}^{p-1} \sum_{k=0}^{a-1} \binom{d}{k} (ip)^k j^{d-k} \\ &= \sum_{k=0}^{a-1} \binom{d}{k} p^k \sum_{j=0}^{p-1} j^{d-k} \sum_{i=0}^{q-1} i^k\end{aligned}$$

- $\sum_{j=0}^{p-1} j^{d-k}$, $\sum_{i=0}^{q-1} i^k$ 两个互相独立，可以分开求和。
- $\sum_{i=0}^{q-1} i^k$ 我们可以递归计算， $\sum_{j=0}^{p-1} j^{d-k}$ 可以暴力预处理。

- 也就是

$$\begin{aligned}\sum_{i=0}^{qp-1} i^d &= \sum_{i=0}^{q-1} \sum_{j=0}^{p-1} \sum_{k=0}^{a-1} \binom{d}{k} (ip)^k j^{d-k} \\ &= \sum_{k=0}^{a-1} \binom{d}{k} p^k \sum_{j=0}^{p-1} j^{d-k} \sum_{i=0}^{q-1} i^k\end{aligned}$$

- $\sum_{j=0}^{p-1} j^{d-k}$, $\sum_{i=0}^{q-1} i^k$ 两个互相独立，可以分开求和。
- $\sum_{i=0}^{q-1} i^k$ 我们可以递归计算， $\sum_{j=0}^{p-1} j^{d-k}$ 可以暴力预处理。
- 对于每个 p ，时间复杂度为 $O(p \log_p^2 m \log d + \log_p^3 m)$ 。

FZU A math problem

FZU A math problem

- 找到一个最小的 N ，对所有的 i 从0到 k 满足 $N \equiv b_i \pmod{P_i^{C_i}}$

FZU A math problem

- 找到一个最小的 N ，对所有的 i 从0到 k 满足 $N \equiv b_i \pmod{P_i^{C_i}}$
- $M = \prod_{i=0}^k p_i^{c_i}$

FZU A math problem

- 找到一个最小的 N ，对所有的 i 从0到 k 满足 $N \equiv b_i \pmod{P_i^{C_i}}$
- $M = \prod_{i=0}^k p_i^{C_i}$
- 求 $Ans = \sum_{i=1}^N i^A \bmod M$

FZU A math problem

- 找到一个最小的 N ，对所有的 i 从0到 k 满足 $N \equiv b_i \pmod{P_i^{C_i}}$
- $M = \prod_{i=0}^k p_i^{C_i}$
- 求 $Ans = \sum_{i=1}^N i^A \pmod M$
- 保证 $50 \leq A \leq 10^9, k \leq 20, 0 \leq b_i \leq 200, N, M \leq 10^9$

FZU A math problem

- 找到一个最小的 N ，对所有的 i 从0到 k 满足 $N \equiv b_i \pmod{P_i^{C_i}}$
- $M = \prod_{i=0}^k p_i^{C_i}$
- 求 $Ans = \sum_{i=1}^N i^A \bmod M$
- 保证 $50 \leq A \leq 10^9, k \leq 20, 0 \leq b_i \leq 200, N, M \leq 10^9$
- Orz AekdyCoin大神

- 用中国剩余定理计算出 N

- 用中国剩余定理计算出 N
- 分别求 Ans 对 $P_0^{C_0}, P_1^{C_1}, \dots, P_k^{C_k}$ 取模的余数，然后用中国剩余定理合并即可。

- 用中国剩余定理计算出 N
- 分别求 Ans 对 $P_0^{C_0}, P_1^{C_1}, \dots, P_k^{C_k}$ 取模的余数，然后用中国剩余定理合并即可。
- 因为 $b_i \leq 200$ ，所以多出的部分可以暴力计算。

- 用中国剩余定理计算出 N
- 分别求 Ans 对 $P_0^{C_0}, P_1^{C_1}, \dots, P_k^{C_k}$ 取模的余数，然后用中国剩余定理合并即可。
- 因为 $b_i \leq 200$ ，所以多出的部分可以暴力计算。
- 问题就转化成了 $\sum_{i=0}^{P^C-1} i^d$ 对 P^C 取模。

- 注意 $d \geq 50 > \log(10^9)$, 那么就有区间内满足 $P|x$ 都可以忽略。

- 注意 $d \geq 50 > \log(10^9)$ ，那么就有区间内满足 $P|x$ 都可以忽略。
- 令 $p = P^C$, g 为 p 的原根。当 i 取遍 0 到 $\varphi(p) - 1$ 时, g^i 取遍 $[0, p)$ 中与 p 互质的数。

- 注意 $d \geq 50 > \log(10^9)$, 那么就有区间内满足 $P|x$ 都可以忽略。
- 令 $p = P^C$, g 为 p 的原根。当 i 取遍 0 到 $\varphi(p) - 1$ 时, g^i 取遍 $[0, p)$ 中与 p 互质的数。

•

$$\sum_{i=0}^{p-1} i^d \equiv \sum_{i=0}^{\varphi(p)-1} g^{id} \pmod{p}$$

- 注意 $d \geq 50 > \log(10^9)$ ，那么就有区间内满足 $P|x$ 都可以忽略。
- 令 $p = P^C$, g 为 p 的原根。当 i 取遍 0 到 $\varphi(p) - 1$ 时, g^i 取遍 $[0, p)$ 中与 p 互质的数。

•

$$\sum_{i=0}^{p-1} i^d \equiv \sum_{i=0}^{\varphi(p)-1} g^{id} \pmod{p}$$

- 这是一个等比数列求和，可以二分计算。

- 当 $P = 2$ 且 $C \geq 3$ 时，原根并不存在。

- 当 $P = 2$ 且 $C \geq 3$ 时, 原根并不存在。
- 令 $S_c = \sum_{i=0}^{2^c-1} i^d$

- 当 $P = 2$ 且 $C \geq 3$ 时, 原根并不存在。
- 令 $S_c = \sum_{i=0}^{2^c-1} i^d$
- 当 d 为奇数时, 有 $k^d + (2^c - k)^d \equiv 0 \pmod{2^c}$, 也就是 $S_c \equiv 0 \pmod{2^c}$

- 当 $P = 2$ 且 $C \geq 3$ 时, 原根并不存在。
- 令 $S_c = \sum_{i=0}^{2^c-1} i^d$
- 当 d 为奇数时, 有 $k^d + (2^c - k)^d \equiv 0 \pmod{2^c}$, 也就是 $S_c \equiv 0 \pmod{2^c}$
- 当 d 为偶数时, 有 $S_c \equiv 2^{c-1} \pmod{2^c}$

- 当 $P = 2$ 且 $C \geq 3$ 时, 原根并不存在。
- 令 $S_c = \sum_{i=0}^{2^c-1} i^d$
- 当 d 为奇数时, 有 $k^d + (2^c - k)^d \equiv 0 \pmod{2^c}$, 也就是 $S_c \equiv 0 \pmod{2^c}$
- 当 d 为偶数时, 有 $S_c \equiv 2^{c-1} \pmod{2^c}$
- 当 $c = 0$ 时 $S_c = 1$, 显然成立

- 当 $P = 2$ 且 $C \geq 3$ 时，原根并不存在。
- 令 $S_c = \sum_{i=0}^{2^c-1} i^d$
- 当 d 为奇数时，有 $k^d + (2^c - k)^d \equiv 0 \pmod{2^c}$ ，也就是 $S_c \equiv 0 \pmod{2^c}$
- 当 d 为偶数时，有 $S_c \equiv 2^{c-1} \pmod{2^c}$
- 当 $c = 0$ 时 $S_c = 1$ ，显然成立
- 当 $c = i$ 时，假设成立

- 当 $P = 2$ 且 $C \geq 3$ 时，原根并不存在。
- 令 $S_c = \sum_{i=0}^{2^c-1} i^d$
- 当 d 为奇数时，有 $k^d + (2^c - k)^d \equiv 0 \pmod{2^c}$ ，也就是 $S_c \equiv 0 \pmod{2^c}$
- 当 d 为偶数时，有 $S_c \equiv 2^{c-1} \pmod{2^c}$
- 当 $c = 0$ 时 $S_c = 1$ ，显然成立
- 当 $c = i$ 时，假设成立
- 那么就有当 $c = i + 1$ 时， $k^d + (2^c - k)^d \equiv 2k^d \pmod{2^c}$ ，也就是 $S_c \equiv 2S_{c-1} \pmod{2^c}$

- 当 $P = 2$ 且 $C \geq 3$ 时, 原根并不存在。
- 令 $S_c = \sum_{i=0}^{2^c-1} i^d$
- 当 d 为奇数时, 有 $k^d + (2^c - k)^d \equiv 0 \pmod{2^c}$, 也就是 $S_c \equiv 0 \pmod{2^c}$
- 当 d 为偶数时, 有 $S_c \equiv 2^{c-1} \pmod{2^c}$
- 当 $c = 0$ 时 $S_c = 1$, 显然成立
- 当 $c = i$ 时, 假设成立
- 那么就有当 $c = i + 1$ 时, $k^d + (2^c - k)^d \equiv 2k^d \pmod{2^c}$, 也就是 $S_c \equiv 2S_{c-1} \pmod{2^c}$
- 因为 $S_{c-1} \equiv 2^{c-2} \pmod{2^{c-1}}$, 所以就有 $S_c \equiv 2^{c-1} \pmod{2^c}$

tyvj xlkxc

- 给定 k, a, n, d, p , 已知 $f(x) = \sum_{i=1}^x i^k, g(x) = \sum_{i=1}^x f(x)$,
求 $\sum_{i=0}^n g(a + id) \bmod p$

- 给定 k, a, n, d, p , 已知 $f(x) = \sum_{i=1}^x i^k, g(x) = \sum_{i=1}^x f(x)$,
求 $\sum_{i=0}^n g(a + id) \bmod p$
- $k \leq 123, a, n, d \leq 123456789, p = 1234567891$

- 给定 k, a, n, d, p , 已知 $f(x) = \sum_{i=1}^x i^k, g(x) = \sum_{i=1}^x f(x)$,
求 $\sum_{i=0}^n g(a + id) \bmod p$
- $k \leq 123, a, n, d \leq 123456789, p = 1234567891$
- p 为素数

- 给定 k, a, n, d, p , 已知 $f(x) = \sum_{i=1}^x i^k, g(x) = \sum_{i=1}^x f(x)$,
求 $\sum_{i=0}^n g(a + id) \bmod p$
- $k \leq 123, a, n, d \leq 123456789, p = 1234567891$
- p 为素数
- Orz XLk大神

法1

法1

- 我们可以通过伯努利数在 $O(k^2)$ 的时间内预处理，并在 $O(k)$ 的时间内算出 $f(x)$ 的表达式。令

$$f(x) = \sum_{i=0}^{k+1} a_i x^i$$

法1

- 我们可以通过伯努利数在 $O(k^2)$ 的时间内预处理，并在 $O(k)$ 的时间内算出 $f(x)$ 的表达式。令

$$f(x) = \sum_{i=0}^{k+1} a_i x^i$$

- 那么

$$g(x) = \sum_{j=1}^x \sum_{i=0}^{k+1} a_i j^i = \sum_{i=0}^{k+1} a_i \sum_{j=1}^x j^i$$

法1

- 我们可以通过伯努利数在 $O(k^2)$ 的时间内预处理，并在 $O(k)$ 的时间内算出 $f(x)$ 的表达式。令

$$f(x) = \sum_{i=0}^{k+1} a_i x^i$$

- 那么

$$g(x) = \sum_{j=1}^x \sum_{i=0}^{k+1} a_i j^i = \sum_{i=0}^{k+1} a_i \sum_{j=1}^x j^i$$

- 由于 $\sum_{j=1}^x j^i$ 的系数可以在 $O(k)$ 时间内算出，那么 $g(k)$ 的表达式可以在 $O(k^2)$ 时间内算出。

- 令

$$g(x) = \sum_{i=0}^{k+2} b_i x^i$$

• 令

$$g(x) = \sum_{i=0}^{k+2} b_i x^i$$

•

$$\begin{aligned} \sum_{i=0}^n g(a+id) &= \sum_{i=0}^n \sum_{j=0}^{k+2} b_j (a+id)^j = \sum_{i=0}^n \sum_{j=0}^{k+2} b_j \sum_{k'=0}^j \binom{j}{k'} a^{k'} (id)^{j-k'} \\ &= \sum_{j=0}^{k+2} b_j \sum_{k'=0}^j \binom{j}{k'} a^{k'} d^{j-k'} \sum_{i=0}^n i^{j-k'} \end{aligned}$$

- 令

$$g(x) = \sum_{i=0}^{k+2} b_i x^i$$

-

$$\begin{aligned} \sum_{i=0}^n g(a+id) &= \sum_{i=0}^n \sum_{j=0}^{k+2} b_j (a+id)^j = \sum_{i=0}^n \sum_{j=0}^{k+2} b_j \sum_{k'=0}^j \binom{j}{k'} a^{k'} (id)^{j-k'} \\ &= \sum_{j=0}^{k+2} b_j \sum_{k'=0}^j \binom{j}{k'} a^{k'} d^{j-k'} \sum_{i=0}^n i^{j-k'} \end{aligned}$$

- 将 $\sum_{i=0}^n i^d$ 的 d 相同的项前面的系数并在一起，然后利用伯努利数计算，时间复杂度为 $O(k^2)$ 。

法2

法2

- 我们可以算出 $f(0), f(1), \dots, f(k+3)$ 的值，然后就可以算出 $g(0), g(1), \dots, g(k+3)$ 的值。

法2

- 我们可以算出 $f(0), f(1), \dots, f(k+3)$ 的值，然后就可以算出 $g(0), g(1), \dots, g(k+3)$ 的值。
- 由于 g 是一个次数为 $k+2$ 的多项式，那么我们肯定能在 $O(k)$ 算出 $g(a)$ ，也就是能在 $O(k^2)$ 的时间复杂度内算出 $g(a), g(a+d), g(a+2d), \dots, g(a+(k+3)d)$ 。

法2

- 我们可以算出 $f(0), f(1), \dots, f(k+3)$ 的值，然后就可以算出 $g(0), g(1), \dots, g(k+3)$ 的值。
- 由于 g 是一个次数为 $k+2$ 的多项式，那么我们肯定能在 $O(k)$ 算出 $g(a)$ ，也就是能在 $O(k^2)$ 的时间复杂度内算出 $g(a), g(a+d), g(a+2d), \dots, g(a+(k+3)d)$ 。
- 由于 $S(n) = \sum_{i=0}^n g(a+id)$ 显然是一个关于 n 次数为 $k+3$ 的多项式，我们知道了 $S(0), S(1), \dots, S(k+3)$ 的值，那么就能算出 $S(n)$ 。

法2

- 我们可以算出 $f(0), f(1), \dots, f(k+3)$ 的值，然后就可以算出 $g(0), g(1), \dots, g(k+3)$ 的值。
- 由于 g 是一个次数为 $k+2$ 的多项式，那么我们肯定能在 $O(k)$ 算出 $g(a)$ ，也就是能在 $O(k^2)$ 的时间复杂度内算出 $g(a), g(a+d), g(a+2d), \dots, g(a+(k+3)d)$ 。
- 由于 $S(n) = \sum_{i=0}^n g(a+id)$ 显然是一个关于 n 次数为 $k+3$ 的多项式，我们知道了 $S(0), S(1), \dots, S(k+3)$ 的值，那么就能算出 $S(n)$ 。
- 时间复杂度还是 $O(k^2)$ ，但是你要实现的仅是给定的 $f(0), f(1), \dots, f(d)$ ，计算出 $f(n)$ 即可。

Codechef QPOLYSUM

Codechef QPOLYSUM

- 给定一个次数为 d 的多项式 $P(x)$ ，给出 $P(0), P(1), \dots, P(d) \bmod M$ 的值，求 $\sum_{i=0}^{n-1} P(i)Q^i \bmod M$ 。

Codechef QPOLYSUM

- 给定一个次数为 d 的多项式 $P(x)$, 给出 $P(0), P(1), \dots, P(d) \bmod M$ 的值, 求 $\sum_{i=0}^{n-1} P(i)Q^i \bmod M$ 。
- $n \leq 10^{100000}, d \leq 20000, 0 \leq Q, M \leq 10^{18}$

Codechef QPOLYSUM

- 给定一个次数为 d 的多项式 $P(x)$ ，给出 $P(0), P(1), \dots, P(d) \bmod M$ 的值，求 $\sum_{i=0}^{n-1} P(i)Q^i \bmod M$ 。
- $n \leq 10^{100000}$, $d \leq 20000$, $0 \leq Q, M \leq 10^{18}$
- M 与 $2, 3, \dots, d + 14$ 互质

Codechef QPOLYSUM

- 给定一个次数为 d 的多项式 $P(x)$ ，给出 $P(0), P(1), \dots, P(d) \bmod M$ 的值，求 $\sum_{i=0}^{n-1} P(i)Q^i \bmod M$ 。
- $n \leq 10^{100000}$, $d \leq 20000$, $0 \leq Q, M \leq 10^{18}$
- M 与 $2, 3, \dots, d + 14$ 互质
- 官方题解的复杂度为 $O(D(K \log D + K^2 + \log(M^2 D)) + \log N \cdot \log M)$ ，其中 $k = \log(M^2 d) / \log(2^{31})$

- 特殊处理 $Q = 0, Q = 1$ 的情况, $Q = 0$ 时答案就是 $P(0)$ 。

- 特殊处理 $Q = 0, Q = 1$ 的情况, $Q = 0$ 时答案就是 $P(0)$ 。
- $Q = 1$ 时我们可以先算出 $P(d + 1)$, 令 $S(n) = \sum_{i=0}^n P(i)$, 那么我们已经知道 $S(0), S(1), \dots, S(d + 1)$, 就可以算出 $S(n)$ 的值。

- 特殊处理 $Q = 0, Q = 1$ 的情况, $Q = 0$ 时答案就是 $P(0)$ 。
- $Q = 1$ 时我们可以先算出 $P(d+1)$, 令 $S(n) = \sum_{i=0}^n P(i)$, 那么我们已经知道 $S(0), S(1), \dots, S(d+1)$, 就可以算出 $S(n)$ 的值。
- 考虑 Q 不等于 0 或 1 时的情况, 令

$$G(n) = \sum_{i=0}^{n-1} P(i)Q^i = Q^n F(n) - F(0)$$

- 特殊处理 $Q = 0, Q = 1$ 的情况, $Q = 0$ 时答案就是 $P(0)$ 。
- $Q = 1$ 时我们可以先算出 $P(d+1)$, 令 $S(n) = \sum_{i=0}^n P(i)$, 那么我们已经知道 $S(0), S(1), \dots, S(d+1)$, 就可以算出 $S(n)$ 的值。
- 考虑 Q 不等于 0 或 1 时的情况, 令

$$G(n) = \sum_{i=0}^{n-1} P(i)Q^i = Q^n F(n) - F(0)$$

- 其中 $F(n)$ 次数为 d 的一个多项式, 这个可以通过数学归纳法得到。

- 当 $d = 0$ 时显然成立。

- 当 $d = 0$ 时显然成立。
- 当 $d = k$ 时，假设成立。

- 当 $d = 0$ 时显然成立。
- 当 $d = k$ 时，假设成立。
- 当 $d = k + 1$ 时，令 $S_d(n) = \sum_{i=0}^{n-1} P(i)Q^i$ 。

- 当 $d = 0$ 时显然成立。
- 当 $d = k$ 时，假设成立。
- 当 $d = k + 1$ 时，令 $S_d(n) = \sum_{i=0}^{n-1} P(i)Q^i$ 。
- $QS_d(n) = \sum_{i=0}^{n-1} P(i)Q^{i+1} = \sum_{i=1}^n P(i-1)Q^i$

- 当 $d = 0$ 时显然成立。
- 当 $d = k$ 时，假设成立。
- 当 $d = k + 1$ 时，令 $S_d(n) = \sum_{i=0}^{n-1} P(i)Q^i$ 。
- $QS_d(n) = \sum_{i=0}^{n-1} P(i)Q^{i+1} = \sum_{i=1}^n P(i-1)Q^i$
- $(Q-1)S_d(n) = P(n-1)Q^n + \sum_{i=0}^{n-1} (P(i-1) - P(i))Q^i - P(-1)$

- 当 $d = 0$ 时显然成立。
- 当 $d = k$ 时，假设成立。
- 当 $d = k + 1$ 时，令 $S_d(n) = \sum_{i=0}^{n-1} P(i)Q^i$ 。
- $QS_d(n) = \sum_{i=0}^{n-1} P(i)Q^{i+1} = \sum_{i=1}^n P(i-1)Q^i$
- $(Q-1)S_d(n) = P(n-1)Q^n + \sum_{i=0}^{n-1} (P(i-1) - P(i))Q^i - P(-1)$
- $P(i-1) - P(i)$ 是一个次数为 $d-1$ 的多项式。

- 当 $d = 0$ 时显然成立。
- 当 $d = k$ 时，假设成立。
- 当 $d = k + 1$ 时，令 $S_d(n) = \sum_{i=0}^{n-1} P(i)Q^i$ 。
- $QS_d(n) = \sum_{i=0}^{n-1} P(i)Q^{i+1} = \sum_{i=1}^n P(i-1)Q^i$
- $(Q-1)S_d(n) = P(n-1)Q^n + \sum_{i=0}^{n-1} (P(i-1) - P(i))Q^i - P(-1)$
- $P(i-1) - P(i)$ 是一个次数为 $d-1$ 的多项式。
- 那么 $\sum_{i=0}^{n-1} (P(i-1) - P(i))Q^i$ 一定能被表示成 $Q^n f(x) - f(0)$

- 当 $d = 0$ 时显然成立。
- 当 $d = k$ 时，假设成立。
- 当 $d = k + 1$ 时，令 $S_d(n) = \sum_{i=0}^{n-1} P(i)Q^i$ 。
- $QS_d(n) = \sum_{i=0}^{n-1} P(i)Q^{i+1} = \sum_{i=1}^n P(i-1)Q^i$
- $(Q-1)S_d(n) = P(n-1)Q^n + \sum_{i=0}^{n-1} (P(i-1) - P(i))Q^i - P(-1)$
- $P(i-1) - P(i)$ 是一个次数为 $d-1$ 的多项式。
- 那么 $\sum_{i=0}^{n-1} (P(i-1) - P(i))Q^i$ 一定能被表示成 $Q^n f(x) - f(0)$
- 那么 $S_d(n)$ 一定能被表示成 $Q^n F(n) - c$ ，其中 $F(n) = (f(n) + P(n-1))/(Q-1)$ ， c 为一个常数。

- 当 $d = 0$ 时显然成立。
- 当 $d = k$ 时，假设成立。
- 当 $d = k + 1$ 时，令 $S_d(n) = \sum_{i=0}^{n-1} P(i)Q^i$ 。
- $QS_d(n) = \sum_{i=0}^{n-1} P(i)Q^{i+1} = \sum_{i=1}^n P(i-1)Q^i$
- $(Q-1)S_d(n) = P(n-1)Q^n + \sum_{i=0}^{n-1} (P(i-1) - P(i))Q^i - P(-1)$
- $P(i-1) - P(i)$ 是一个次数为 $d-1$ 的多项式。
- 那么 $\sum_{i=0}^{n-1} (P(i-1) - P(i))Q^i$ 一定能被表示成 $Q^n f(x) - f(0)$
- 那么 $S_d(n)$ 一定能被表示成 $Q^n F(n) - c$ ，其中 $F(n) = (f(n) + P(n-1))/(Q-1)$ ， c 为一个常数。
- 考虑 $n = 0$ 时， $S_d(n) = 0$ ，也就是 $c = F(0)$

- 当 $d = 0$ 时显然成立。
- 当 $d = k$ 时，假设成立。
- 当 $d = k + 1$ 时，令 $S_d(n) = \sum_{i=0}^{n-1} P(i)Q^i$ 。
- $QS_d(n) = \sum_{i=0}^{n-1} P(i)Q^{i+1} = \sum_{i=1}^n P(i-1)Q^i$
- $(Q-1)S_d(n) = P(n-1)Q^n + \sum_{i=0}^{n-1} (P(i-1) - P(i))Q^i - P(-1)$
- $P(i-1) - P(i)$ 是一个次数为 $d-1$ 的多项式。
- 那么 $\sum_{i=0}^{n-1} (P(i-1) - P(i))Q^i$ 一定能被表示成 $Q^n f(x) - f(0)$
- 那么 $S_d(n)$ 一定能被表示成 $Q^n F(n) - c$ ，其中 $F(n) = (f(n) + P(n-1))/(Q-1)$ ， c 为一个常数。
- 考虑 $n = 0$ 时， $S_d(n) = 0$ ，也就是 $c = F(0)$
- $F(n)$ 显然是一个次数为 d 的多项式， $S_d(n) = Q^n F(n) - F(0)$

- $G(n) = \sum_{i=0}^{n-1} P(i)Q^i = Q^n F(n) - F(0)$

- $G(n) = \sum_{i=0}^{n-1} P(i)Q^i = Q^n F(n) - F(0)$
- 相减得 $G(n+1) - G(n) = P(n)Q^n = Q^{n+1}F(n+1) - Q^n F(n)$

- $G(n) = \sum_{i=0}^{n-1} P(i)Q^i = Q^n F(n) - F(0)$
- 相减得 $G(n+1) - G(n) = P(n)Q^n = Q^{n+1}F(n+1) - Q^n F(n)$
- 即

$$P(n) = QF(n+1) - F(n) \Rightarrow F(n+1) = \frac{F(n) + P(n)}{Q}$$

- $G(n) = \sum_{i=0}^{n-1} P(i)Q^i = Q^n F(n) - F(0)$
- 相减得 $G(n+1) - G(n) = P(n)Q^n = Q^{n+1}F(n+1) - Q^n F(n)$
- 即

$$P(n) = QF(n+1) - F(n) \Rightarrow F(n+1) = \frac{F(n) + P(n)}{Q}$$

- 我们并不知道 $F(0)$ 的值，但可以通过递推把 $F(1), F(2), \dots, F(d), F(d+1)$ 表示成关于 $F(0)$ 的一次函数。

- $G(n) = \sum_{i=0}^{n-1} P(i)Q^i = Q^n F(n) - F(0)$
- 相减得 $G(n+1) - G(n) = P(n)Q^n = Q^{n+1}F(n+1) - Q^n F(n)$
- 即

$$P(n) = QF(n+1) - F(n) \Rightarrow F(n+1) = \frac{F(n) + P(n)}{Q}$$

- 我们并不知道 $F(0)$ 的值，但可以通过递推把 $F(1), F(2), \dots, F(d), F(d+1)$ 表示成关于 $F(0)$ 的一次函数。
- 由于 $F(x)$ 是一个次数为 d 的多项式，那么就满足

$$\sum_{i=0}^{d+1} (-1)^i \binom{d+1}{i} F(i) = 0$$

- $G(n) = \sum_{i=0}^{n-1} P(i)Q^i = Q^n F(n) - F(0)$
- 相减得 $G(n+1) - G(n) = P(n)Q^n = Q^{n+1}F(n+1) - Q^n F(n)$
- 即

$$P(n) = QF(n+1) - F(n) \Rightarrow F(n+1) = \frac{F(n) + P(n)}{Q}$$

- 我们并不知道 $F(0)$ 的值，但可以通过递推把 $F(1), F(2), \dots, F(d), F(d+1)$ 表示成关于 $F(0)$ 的一次函数。
- 由于 $F(x)$ 是一个次数为 d 的多项式，那么就满足

$$\sum_{i=0}^{d+1} (-1)^i \binom{d+1}{i} F(i) = 0$$

- 这就是个关于 $F(0)$ 的一次方程，我们可以解出 $F(0)$ 的值。

- 注意到这里在递推和解方程中涉及了除法运算，但在模 M 的条件下不一定有逆元。

- 注意到这里在递推和解方程中涉及了除法运算，但在模 M 的条件下不一定有逆元。
- 递推中涉及到除 Q 。

- 注意到这里在递推和解方程中涉及了除法运算，但在模 M 的条件下不一定有逆元。
- 递推中涉及到除 Q 。
- $F(i)$ 中 $F(0)$ 的系数为 Q^{-i} ，那么最终方程中 $F(0)$ 的系数就为

$$\sum_{i=0}^{d+1} (-1)^i \binom{d+1}{i} Q^{-i} = (1 - Q^{-1})^{d+1}$$

- 注意到这里在递推和解方程中涉及了除法运算，但在模 M 的条件下不一定有逆元。
- 递推中涉及到除 Q 。
- $F(i)$ 中 $F(0)$ 的系数为 Q^{-i} ，那么最终方程中 $F(0)$ 的系数就为

$$\sum_{i=0}^{d+1} (-1)^i \binom{d+1}{i} Q^{-i} = (1 - Q^{-1})^{d+1}$$

- 那么在解方程中涉及到 $(Q - 1)$ 的逆元。

- 令 $M = m_1 m_2 m_3$, 存在 u, v 使得 $(m_2 m_3, Q) = 1, m_1 | Q^u$,
 $(m_1 m_3, Q - 1) = 1, m_2 | (Q - 1)^v$, u, v 是满足条件的最小的值。

- 令 $M = m_1 m_2 m_3$, 存在 u, v 使得 $(m_2 m_3, Q) = 1, m_1 | Q^u$, $(m_1 m_3, Q - 1) = 1, m_2 | (Q - 1)^v$, u, v 是满足条件的最小的值。
- 可以分别算出模 m_1, m_2, m_3 的余数, 然后用中国剩余定理合并。

- 令 $M = m_1 m_2 m_3$, 存在 u, v 使得 $(m_2 m_3, Q) = 1, m_1 | Q^u$, $(m_1 m_3, Q - 1) = 1, m_2 | (Q - 1)^v$, u, v 是满足条件的最小的值。
- 可以分别算出模 m_1, m_2, m_3 的余数, 然后用中国剩余定理合并。
- 对 m_3 取模可以用上述的方法解决

- 令 $M = m_1 m_2 m_3$, 存在 u, v 使得 $(m_2 m_3, Q) = 1, m_1 | Q^u$, $(m_1 m_3, Q - 1) = 1, m_2 | (Q - 1)^v$, u, v 是满足条件的最小的值。
- 可以分别算出模 m_1, m_2, m_3 的余数, 然后用中国剩余定理合并。
- 对 m_3 取模可以用上述的方法解决
- 由于 $m_1 | Q^u$, 当 $i \geq u$ 时, $P(i)Q^i$ 对 Q^u 取模为 0, 可以忽略。

- 令 $m_1 = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$

- 令 $m_1 = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$
- 又因为 $m_1 | Q^u, m_1 \nmid Q^{u-1}$, 这样就有 $u \leq \max\{a_1, a_2, \dots, a_k\}$

- 令 $m_1 = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$
- 又因为 $m_1 | Q^u, m_1 \nmid Q^{u-1}$, 这样就有 $u \leq \max\{a_1, a_2, \dots, a_k\}$
- 又因为 M 和 $2, 3, \dots, d+14$ 的数互质, 那么 M 最小的可能素因子是 17。

- 令 $m_1 = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$
- 又因为 $m_1 | Q^u, m_1 \nmid Q^{u-1}$, 这样就有 $u \leq \max\{a_1, a_2, \dots, a_k\}$
- 又因为 M 和 $2, 3, \dots, d+14$ 的数互质, 那么 M 最小的可能素因子是 17。
- 又有 $17^{15} > 10^{18}$, 那么就有 $a_i \leq 14$, 也就是 $u \leq 14$ 。

- 令 $m_1 = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$
- 又因为 $m_1 | Q^u$, $m_1 \nmid Q^{u-1}$, 这样就有 $u \leq \max\{a_1, a_2, \dots, a_k\}$
- 又因为 M 和 $2, 3, \dots, d+14$ 的数互质, 那么 M 最小的可能素因子是 17。
- 又有 $17^{15} > 10^{18}$, 那么就有 $a_i \leq 14$, 也就是 $u \leq 14$ 。
- 同理的 $v \leq 14$, 即 M 和 $2, 3, \dots, d+v$ 互质。

- 对 m_2 取模, 有

$$P(i)Q^i = P(i)((Q-1)+1)^i = P(i) \sum_{j=0}^i \binom{i}{j} (Q-1)^j$$

- 对 m_2 取模, 有

$$P(i)Q^i = P(i)((Q-1)+1)^i = P(i) \sum_{j=0}^i \binom{i}{j} (Q-1)^j$$

- 当 $j \geq v$ 时, $(Q-1)^j$ 对 $(Q-1)^v$ 取模后值为 0, 可以忽略。

- 对 m_2 取模, 有

$$P(i)Q^i = P(i)((Q-1)+1)^i = P(i) \sum_{j=0}^i \binom{i}{j} (Q-1)^j$$

- 当 $j \geq v$ 时, $(Q-1)^j$ 对 $(Q-1)^v$ 取模后值为 0, 可以忽略。



$$\sum_{i=0}^{n-1} P(i)Q^i = \sum_{j=0}^{v-1} (Q-1)^j \sum_{i=0}^{n-1} P(i) \binom{i}{j}$$

- 对 m_2 取模, 有

$$P(i)Q^i = P(i)((Q-1)+1)^i = P(i) \sum_{j=0}^i \binom{i}{j} (Q-1)^j$$

- 当 $j \geq v$ 时, $(Q-1)^j$ 对 $(Q-1)^v$ 取模后值为 0, 可以忽略。

-

$$\sum_{i=0}^{n-1} P(i)Q^i = \sum_{j=0}^{v-1} (Q-1)^j \sum_{i=0}^{n-1} P(i) \binom{i}{j}$$

- $\binom{i}{j}$ 显然是关于 i 的多项式, 那么 $P(i)\binom{i}{j}$ 是一个次数不超过 $d + v - 1$ 的多项式。

- 对 m_2 取模, 有

$$P(i)Q^i = P(i)((Q-1)+1)^i = P(i) \sum_{j=0}^i \binom{i}{j} (Q-1)^j$$

- 当 $j \geq v$ 时, $(Q-1)^j$ 对 $(Q-1)^v$ 取模后值为 0, 可以忽略。

-

$$\sum_{i=0}^{n-1} P(i)Q^i = \sum_{j=0}^{v-1} (Q-1)^j \sum_{i=0}^{n-1} P(i) \binom{i}{j}$$

- $\binom{i}{j}$ 显然是关于 i 的多项式, 那么 $P(i)\binom{i}{j}$ 是一个次数不超过 $d+v-1$ 的多项式。
- 也就是说 $\sum_{i=0}^{n-1} P(i)Q^i$ 是一个关于 n 的次数不超过 $d+v$ 的多项式。

- 令 $s(n) = \sum_{i=0}^{n-1} P(i)Q^i \bmod m_2$, $s(n)$ 是一个关于 n 的多项式。

- 令 $s(n) = \sum_{i=0}^{n-1} P(i)Q^i \bmod m_2$, $s(n)$ 是一个关于 n 的多项式。
- 我们可以算出 $s(0), s(1), \dots, s(d + v)$

- 令 $s(n) = \sum_{i=0}^{n-1} P(i)Q^i \bmod m_2$, $s(n)$ 是一个关于 n 的多项式。
- 我们可以算出 $s(0), s(1), \dots, s(d+v)$
- 因为 m_2 和 $2, 3, \dots, d+v$ 互质, 那么我们就可以算出 $s(n)$

- 令 $s(n) = \sum_{i=0}^{n-1} P(i)Q^i \bmod m_2$, $s(n)$ 是一个关于 n 的多项式。
- 我们可以算出 $s(0), s(1), \dots, s(d+v)$
- 因为 m_2 和 $2, 3, \dots, d+v$ 互质, 那么我们就可以算出 $s(n)$
- 时间复杂度为 $O(d + \log n)$, 这种做法效率高, 代码简单。

- 感谢长郡中学罗雨屏同学对我的帮助

- 感谢长郡中学罗雨屏同学对我的帮助
- 谢谢大家的倾听

- 感谢长郡中学罗雨屏同学对我的帮助
- 谢谢大家的倾听
- 欢迎交流和指出错误