OI 中的数学方法

Wearry

Oct 2nd, 2018

Table of Contents

Table of Contents

Table of Contents

gcd 的一些性质

1. $$\gcd(x^a - 1, x^b - 1) = x^{\gcd(a,b)} - 1$$
2. $$\gcd(fib_a, fib_b) = fib_{\gcd(a,b)}$$

欧拉定理及其拓展

如果有 $\gcd(a,p)=1$:

$$a^{\varphi(p)} \equiv 1 \pmod p$$

一般情况:

$$a^t \equiv a^{\min(t,\, t \bmod \varphi(p)+\varphi(p))} \pmod p$$

同余方程

形如:

$$x \equiv a_i \pmod{p_i}$$

同余方程

形如:

$$x \equiv a_i \pmod{p_i}$$

中国剩余定理

若 p_i 两两互质，存在通解:

$$\begin{aligned} P &= \prod_i p_i \\ P_i &= \frac{P}{p_i} \\ T_i &= P_i^{-1} \bmod p_i \\ x &\equiv \sum_i a_i T_i P_i \pmod{P} \end{aligned}$$

拓展形式

考虑合并两个方程:

$$
\begin{aligned}
x &\equiv a_1 \pmod{p_1} \\
x &\equiv a_2 \pmod{p_2} \\
x &= a_1 + k_1 p_1 = a_2 + k_2 p_2
\end{aligned}
$$

拓展形式

考虑合并两个方程:

$$
\begin{aligned}
x &\equiv a_1 \pmod{p_1} \\
x &\equiv a_2 \pmod{p_2} \\
x &= a_1 + k_1 p_1 = a_2 + k_2 p_2
\end{aligned}
$$

令 $t = \gcd(p_1, p_2)$，则:

$$
\begin{aligned}
k_1 \frac{p_1}{t} &\equiv \frac{a_2 - a_1}{t} \pmod{\frac{p_2}{t}} \\
k_1 &\equiv \frac{a_2 - a_1}{t} \times (\frac{p_1}{t})^{-1} \pmod{\frac{p_2}{t}} \\
x &\equiv a_1 + p_1 \left(\frac{a_2 - a_1}{t} \times (\frac{p_1}{t})^{-1} \bmod \frac{p_2}{t} \right) \pmod{\frac{p_1 p_2}{t}}
\end{aligned}
$$

积性函数

令 $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$:

积性函数

令 $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$:

- $\epsilon(n) = [n = 1]$
- $1(n) = 1$
- $\mathrm{Id}(n) = n$

积性函数

令 $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$:

- $\epsilon(n) = [n = 1]$
- $1(n) = 1$
- $\mathrm{Id}(n) = n$
- $\mu(n) = [\max(e_1, e_2, \cdots, e_k) \leq 1](-1)^k$
- $\varphi(n) = n \prod_{i=1}^{k}(1 - \frac{1}{p_i})$

积性函数

令 $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$:

- $\epsilon(n) = [n = 1]$
- $1(n) = 1$
- $\mathrm{Id}(n) = n$
- $\mu(n) = [\max(e_1, e_2, \cdots, e_k) \leq 1](-1)^k$
- $\varphi(n) = n \prod_{i=1}^{k} (1 - \frac{1}{p_i})$
- $\mathrm{d}(n) = \sum_{d|n} 1$
- $\sigma(n) = \sum_{d|n} d$
- $\lambda(n) = (-1)^k$

Dirichlet 卷积

$$(f * g)(n) = \sum_{d|n} f(d)g(\frac{n}{d})$$

Dirichlet 卷积

$$(f * g)(n) = \sum_{d|n} f(d)g(\frac{n}{d})$$

- $\mu * 1 = \epsilon$
- $\mathrm{Id} = \varphi * 1 \Rightarrow \varphi = \mathrm{Id} * \mu$
- $\mathrm{d} = 1 * 1 \Rightarrow 1 = \mu * \mathrm{d}$
- $\sigma = \mathrm{Id} * 1 \Rightarrow \mathrm{Id} = \mu * \sigma \Rightarrow \sigma = \varphi * \mathrm{d}$

Dirichlet 卷积

$$(f * g)(n) = \sum_{d|n} f(d)g(\frac{n}{d})$$

- $\mu * 1 = \epsilon$
- $\mathrm{Id} = \varphi * 1 \Rightarrow \varphi = \mathrm{Id} * \mu$
- $\mathrm{d} = 1 * 1 \Rightarrow 1 = \mu * \mathrm{d}$
- $\sigma = \mathrm{Id} * 1 \Rightarrow \mathrm{Id} = \mu * \sigma \Rightarrow \sigma = \varphi * \mathrm{d}$
- $d(ij) = \sum_{x|i} \sum_{y|j} [\gcd(x, y) = 1]$
- $\sigma(ij) = \sum_{x|i} \sum_{y|j} [\gcd(x, y) = 1] \frac{iy}{x}$

满足交换律, 结合律, 两个积性函数的卷积也是积性的:

组合数的计算

计算:

$$\binom{n}{m} \bmod p$$

组合数的计算

计算:

$$\binom{n}{m} \bmod p$$

- $n, m \leq 5000$

组合数的计算

计算:

$$\binom{n}{m} \bmod p$$

- $n, m \leq 5000$
- $n, m \leq 10^6, p$ 是质数

组合数的计算

计算:

$$\binom{n}{m} \bmod p$$

- $n, m \le 5000$
- $n, m \le 10^6, p$ 是质数
- $n, m \le 10^{18}, p \le 10^6, p$ 是质数

组合数的计算

计算:

$$\binom{n}{m} \bmod p$$

- $n, m \le 5000$
- $n, m \le 10^6, p$ 是质数
- $n, m \le 10^{18}, p \le 10^6, p$ 是质数
- $n, m \le 10^{18}, p \le 10^6$.

基本组合恒等式

$$\sum_{i=0}^{n} \binom{n}{i} = 2^n$$

$$\sum_{i=0}^{n} \binom{i}{x} = \binom{n+1}{x+1}$$

$$\sum_{i=0}^{n} \binom{k+i}{i} = \binom{k+n+1}{n}$$

$$\sum_{i=0}^{m} \binom{m}{i} \binom{n-m}{m-i} = \binom{n}{m}$$

第一类 Stirling 数

1. **定义**: $\begin{bmatrix} n \\ m \end{bmatrix}$ 表示将 n 个物品分为 m 个无序非空环的方案数.

第一类 Stirling 数

1. 定义: $\begin{bmatrix} n \\ m \end{bmatrix}$ 表示将 n 个物品分为 m 个无序非空环的方案数.
2. 递推式:

$$\begin{bmatrix} n \\ m \end{bmatrix} = \begin{bmatrix} n-1 \\ m-1 \end{bmatrix} + (n-1) \begin{bmatrix} n-1 \\ m \end{bmatrix}$$

第一类 Stirling 数

1. 定义: $\begin{bmatrix} n \\ m \end{bmatrix}$ 表示将 n 个物品分为 m 个无序非空环的方案数.
2. 递推式:

$$\begin{bmatrix} n \\ m \end{bmatrix} = \begin{bmatrix} n-1 \\ m-1 \end{bmatrix} + (n-1) \begin{bmatrix} n-1 \\ m \end{bmatrix}$$

3. 生成函数:

$$x^{\overline{n}} = \sum_{i=0}^{n} \begin{bmatrix} n \\ i \end{bmatrix} x^i$$
$$x^{\underline{n}} = \sum_{i=0}^{n} (-1)^{n-i} \begin{bmatrix} n \\ i \end{bmatrix} x^i$$

第二类 Stirling 数

1. 定义: $\left\{ {n \atop m} \right\}$ 表示将 n 个物品分成 m 个无序非空集合的方案数.

第二类 Stirling 数

1. **定义**: $\left\{ n \atop m \right\}$ **表示将** n **个物品分成** m **个无序非空集合的方案数**.
2. **递推式**:

$$\left\{ n \atop m \right\} = \left\{ n-1 \atop m-1 \right\} + m\left\{ n-1 \atop m \right\}$$

第二类 Stirling 数

1. 定义: $\left\{ n \atop m \right\}$ 表示将 n 个物品分成 m 个无序非空集合的方案数.
2. 递推式:

$$\left\{ n \atop m \right\} = \left\{ n-1 \atop m-1 \right\} + m \left\{ n-1 \atop m \right\}$$

3. 生成函数:

$$x^n = \sum_{i=0}^{n} \left\{ n \atop i \right\} x^{\underline{i}}$$

$$m! \left\{ n \atop m \right\} = \sum_{i=0}^{m} (-1)^{m-i} \binom{m}{i} i^n$$

多项式的差分序列

定义 $\Delta^k f(n)$ **为** $f(n)$ **的** k **阶差分序列, 并且**:

$$\Delta^k f(n) = \begin{cases} f(n), & \text{k} = 0 \\ \Delta^{k-1} f(n+1) - \Delta^{k-1} f(n), & \text{otherwise} \end{cases}$$

多项式的差分序列

定义 $\Delta^k f(n)$ 为 $f(n)$ 的 k 阶差分序列, 并且:

$$\Delta^k f(n) = \begin{cases} f(n), & \text{k} = 0 \\ \Delta^{k-1} f(n+1) - \Delta^{k-1} f(n), & \text{otherwise} \end{cases}$$

容易发现差分序列的具有线性性, 特别地, 多项式:

$$f(x) = \{0, 0, 0, \cdots, 1\} \mid x \in [0, n]$$

差分序列的第一列是 $\{0, 0, 0, \cdots, 1\}$，事实上这个多项式就是

$$\frac{1}{n!} \prod_{i=0}^{n-1} (x - i) = \binom{x}{n}$$

.

多项式插值

已知一个 n 次多项式的 $n+1$ 个点值, 求这个多项式的系数表示.

牛顿插值

求出多项式的 n 阶差分序列第一列 $\{c_i\}$, 可以将多项式表示成:

$$f(x) = \sum_{i=0}^{n} c_i \binom{x}{i}$$

多项式插值

已知一个 n 次多项式的 $n+1$ 个点值, 求这个多项式的系数表示.

牛顿插值

求出多项式的 n 阶差分序列第一列 $\{c_i\}$, 可以将多项式表示成:

$$f(x)=\sum_{i=0}^{n} c_i\binom{x}{i}$$

拉格朗日插值

考虑构造一个经过所有给定点的多项式:

$$g_i(x)=\prod_{j=0,j\neq i}^{n}\frac{x-x_j}{x_i-x_j}$$

$$f(x)=\sum_{i=0}^{n} y_i g_i(x)$$

自然数幂和

求:

$$f_k(n) = \sum_{i=0}^{n} i^k$$

自然数幂和

求:

$$f_k(n)=\sum_{i=0}^{n} i^k$$

Stirling 数

直接用第二类斯特林数展开:

$$\begin{aligned} f_k(n) &= \sum_{i=0}^{n}\sum_{j=0}^{k}\begin{Bmatrix}k\\j\end{Bmatrix} i^{\underline{j}} \\ &= \sum_{j=0}^{k}\begin{Bmatrix}k\\j\end{Bmatrix} j!\sum_{i=0}^{n}\binom{i}{j} \\ &= \sum_{j=0}^{k}\begin{Bmatrix}k\\j\end{Bmatrix} j!\binom{n+1}{j+1} \end{aligned}$$

自然数幂和

拉格朗日插值

通过上一个方法我们知道 $f_k(n)$ 是一个关于 n 的 $k+1$ 次多项式，于是直接拉格朗日插值即可，并且由于系数的特殊性质，可以做到 $O(k)$.

自然数幂和

拉格朗日插值

通过上一个方法我们知道 $f_k(n)$ 是一个关于 n 的 $k+1$ 次多项式，于是直接拉格朗日插值即可，并且由于系数的特殊性质，可以做到 $O(k)$.

Bernoulli 数

定义 B_i 为伯努利数，满足:

$$\sum_{i=0}^{m} B_i \binom{m+1}{i} = [m=0]$$

那么有:

$$f_k(n-1) = \frac{1}{k+1} \sum_{i=0}^{k} B_i n^{k+1-i} \binom{k+1}{i}$$

容斥与反演

1. $\mathrm{Min}-\mathrm{Max}$ **容斥**:

$$\max(S) = \sum_{T \subseteq S, T \neq \varnothing} \min(T)^{(-1)^{|T|-1}}$$

容斥与反演

1. $\text{Min} - \text{Max}$ 容斥:

$$\max(S) = \sum_{T \subseteq S, T \neq \varnothing} \min(T)^{(-1)^{|T|-1}}$$

2. 拓展形式:

$$\text{lcm}(S) = \prod_{T \subseteq S, T \neq \varnothing} \gcd(T)^{(-1)^{|T|-1}}$$

容斥与反演

3. **二项式反演:**

$$
\begin{aligned}
f(n) &= \sum_{i=0}^{n} \binom{n}{i} g(i) \\
\Leftrightarrow g(n) &= \sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} f(i)
\end{aligned}
$$

容斥与反演

3. 二项式反演:

$$f(n)=\sum_{i=0}^{n}\binom{n}{i}g(i)$$
$$\Leftrightarrow g(n)=\sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}f(i)$$

4. Stirling 反演

$$f(n)=\sum_{i=0}^{n}\left\{ n \atop i \right\}g(i)$$
$$\Leftrightarrow g(n)=\sum_{i=0}^{n}(-1)^{n-i}\left[n \atop i \right]f(i)$$

BZOJ4833

已知:

$$f(n) = \begin{cases} 0, & \text{n} = 0 \\ 1, & \text{n} = 1 \\ 2f(n-1) + f(n-2), & \text{otherwise} \end{cases}$$

$$g(n) = \operatorname{lcm}(f(1), f(2), \cdots, f(n))$$

求:

$$\sum_{i=1}^{n} g(i) \times i$$

$n \leq 10^6$

首先有 $\gcd(f(i), f(j)) = f(\gcd(i, j))$
根据 $\mathrm{Min} - \mathrm{Max}$ 容斥有:

$$\begin{aligned} g(n) &= \prod_{T \subseteq S, T \neq \varnothing} \gcd(T)^{(-1)^{|T|+1}} \\ &= \prod_{T \subseteq S, T \neq \varnothing} f(\gcd(T))^{(-1)^{|T|+1}} \end{aligned}$$

$$\begin{aligned} f(n) &= \prod_{d|n} h(d) \\ \Rightarrow g(n) &= \prod_{d=1}^{n} h(d)^{\sum_{T \subseteq S, T \neq \varnothing} [d|gcd(T)](-1)^{|T|+1}} \\ &= \prod_{d=1}^{n} h(d) \end{aligned}$$

Square

给出一个 $n \times m$ 大小的矩形，每个位置可以填上 $[1, c]$ 中的任意一个数，要求填好后任意两行互不等价且任意两列互不等价，两行或两列等价当且仅当对应位置完全相同，求方案数.

$n, m \leq 5000$

Square

首先我们有一个很简单的方式使得列之间互不等价，对于任意一列，总方案数是 c^n，那么使得列与列之间互不相同的方案数为 $(c^n)^{\underline{m}}$.

Square

首先我们有一个很简单的方式使得列之间互不等价, 对于任意一列, 总方案数是 c^n, 那么使得列与列之间互不相同的方案数为 $(c^n)^{\underline{m}}$.

接下来的问题只与行数有关, 定义 $g(n)$ 表示 n 行不保证每行互不等价的方案数, $f(n)$ 表示 n 行保证任意两行互不等价的方案数, 有:

Square

首先我们有一个很简单的方式使得列之间互不等价，对于任意一列，总方案数是 c^n，那么使得列与列之间互不相同的方案数为 $(c^n)^{\underline{m}}$.

接下来的问题只与行数有关，定义 $g(n)$ 表示 n 行不保证每行互不等价的方案数，$f(n)$ 表示 n 行保证任意两行互不等价的方案数，有：

$$\begin{aligned} g(n) &= (c^n)^{\underline{m}} \\ &= \sum_{i=0}^{n} \begin{Bmatrix} n \\ i \end{Bmatrix} f(i) \\ f(n) &= \sum_{i=0}^{n} (-1)^{n-i} \begin{bmatrix} n \\ i \end{bmatrix} g(i) \end{aligned}$$

Sequence

给出一个长度为 n 的序列 $\{a_i\}$ 以及一个数 p, 现在有 m 次操作, 每次操作将 $[l,r]$ 区间内的 a_i 变成 c^{a_i}, 或者询问 $[l,r]$ 之间所有 a_i 的和对 p 取模的结果.

$n,m \le 5 \times 10^4, p \le 2^{14}$

Sequence

对于修改操作可以利用拓展欧拉定理, 维护一个 $\log(p)$ 层的结构表示每一个 a_i 的值.

Sequence

对于修改操作可以利用拓展欧拉定理, 维护一个 $\log(p)$ 层的结构表示每一个 a_i 的值.

由于经过只有最后的 $\log(p)$ 次操作是有效的, 所以任意的两个相邻位置在经过 $\log(p)$ 次操作后会变得等价, 在最外层维护一个 `std::set` 记录等价的区间, 然后用线段树做询问.

Sequence

对于修改操作可以利用拓展欧拉定理, 维护一个 $\log(p)$ 层的结构表示每一个 a_i 的值.

由于经过只有最后的 $\log(p)$ 次操作是有效的, 所以任意的两个相邻位置在经过 $\log(p)$ 次操作后会变得等价, 在最外层维护一个 `std::set` 记录等价的区间, 然后用线段树做询问.

复杂度的正确性可以通过势能函数简单地分析.