多项式

任轩笛

EECS,PKU

2019 年 6 月 27 日

Contents

Contents

Contents

n 次单位根

n 次单位根

n 次单位根是 $x^n = 1$ 在复数域上的 n 个不同的根，是复平面上单位圆盘的 n 等分点。记为 $1, \omega_n, \omega_n^2 ... \omega_n^{n-1}$。

n 次单位根

n 次单位根是 $x^n = 1$ 在复数域上的 n 个不同的根，是复平面上单位圆盘的 n 等分点。记为 $1, \omega_n, \omega_n^2 ... \omega_n^{n-1}$。

除 1 外，其它 $n-1$ 个 n 次单位根也是 $x^{n-1} + x^{n-2} + ... + 1 = 0 \left(\frac{x^n - 1}{x - 1} = 0 \right)$ 的根。

n 次单位根

n 次单位根是 $x^n = 1$ 在复数域上的 n 个不同的根，是复平面上单位圆盘的 n 等分点。记为 $1, \omega_n, \omega_n^2 ... \omega_n^{n-1}$。

除 1 外，其它 $n-1$ 个 n 次单位根也是 $x^{n-1} + x^{n-2} + ... + 1 = 0 \left(\frac{x^n - 1}{x - 1} = 0 \right)$ 的根。而 1 带进左边的式子得 n。

n 次单位根

n 次单位根是 $x^n = 1$ 在复数域上的 n 个不同的根，是复平面上单位圆盘的 n 等分点。记为 $1, \omega_n, \omega_n^2 ... \omega_n^{n-1}$。

除 1 外，其它 $n-1$ 个 n 次单位根也是 $x^{n-1} + x^{n-2} + ... + 1 = 0 \left(\frac{x^n - 1}{x - 1} = 0 \right)$ 的根。而 1 带进左边的式子得 n。

$\omega_n = e^{\frac{2\pi i}{n}} = \cos \frac{2\pi}{n} + i \sin \frac{2\pi}{n}$ （$e^{iu} = \cos u + i \sin u$）。

单位根的性质

单位根的性质

- $\omega_n^a \omega_n^b = \omega_n^{a+b}$.

单位根的性质

- $\omega_n^a \omega_n^b = \omega_n^{a+b}$.
- 若 $a \equiv b \pmod{n}$ 则 $\omega_n^a = \omega_n^b$.

单位根的性质

- $\omega_n^a \omega_n^b = \omega_n^{a+b}$.
- 若 $a \equiv b \pmod{n}$ 则 $\omega_n^a = \omega_n^b$.
- $\omega_{dn}^{dk} = \omega_n^k$

单位根的性质

- $\omega_n^a \omega_n^b = \omega_n^{a+b}$.
- 若 $a \equiv b \pmod{n}$ 则 $\omega_n^a = \omega_n^b$.
- $\omega_{dn}^{dk} = \omega_n^k \Rightarrow \omega_{2n}^n = \omega_2 = -1$

单位根的性质

- $\omega_n^a\omega_n^b = \omega_n^{a+b}$.
- 若 $a \equiv b \pmod{n}$ 则 $\omega_n^a = \omega_n^b$.
- $\omega_{dn}^{dk} = \omega_n^k \Rightarrow \omega_{2n}^n = \omega_2 = -1 \Rightarrow \omega_{2n}^{k+n} = -\omega_{2n}^k$.

单位根的性质

- $\omega_n^a \omega_n^b = \omega_n^{a+b}$.
- 若 $a \equiv b \pmod{n}$ 则 $\omega_n^a = \omega_n^b$.
- $\omega_{dn}^{dk} = \omega_n^k \Rightarrow \omega_{2n}^n = \omega_2 = -1 \Rightarrow \omega_{2n}^{k+n} = -\omega_{2n}^k$.
- $\overline{\omega_n} = \cos\frac{2\pi}{n} - i\sin\frac{2\pi}{n} = \omega_n^{-1}$.

单位根的性质

- $\omega_n^a \omega_n^b = \omega_n^{a+b}$.
- 若 $a \equiv b \pmod{n}$ 则 $\omega_n^a = \omega_n^b$.
- $\omega_{dn}^{dk} = \omega_n^k \Rightarrow \omega_{2n}^n = \omega_2 = -1 \Rightarrow \omega_{2n}^{k+n} = -\omega_{2n}^k$.
- $\overline{\omega_n} = \cos\frac{2\pi}{n} - i\sin\frac{2\pi}{n} = \omega_n^{-1}$.
- 全部 n 次单位根和为 0。

傅立叶矩阵

傅立叶矩阵

$$F_n = [\omega_n^{ij}]_{0 \le i,j < n} (\omega_n = e^{\frac{2\pi i}{n}})$$

傅立叶矩阵

$$F_n = [\omega_n^{ij}]_{0\le i,j<n}(\omega_n = e^{\frac{2\pi i}{n}})$$

$$= \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \dots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \dots & \omega_n^{2(n-1)} \\ & & \dots & & \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)^2} \end{bmatrix}$$

傅立叶矩阵

$$F_n = [\omega_n^{ij}]_{0 \le i,j < n} (\omega_n = e^{\frac{2\pi i}{n}})$$

$$= \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \dots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \dots & \omega_n^{2(n-1)} \\ & & \dots & & \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)^2} \end{bmatrix}$$

$F_n \overline{F_n} = \overline{F_n} F_n = nI_n$。

傅立叶矩阵

$$F_n = [\omega_n^{ij}]_{0 \le i,j < n} (\omega_n = e^{\frac{2\pi i}{n}})$$

$$= \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \dots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \dots & \omega_n^{2(n-1)} \\ & & \dots & & \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)^2} \end{bmatrix}$$

$F_n\overline{F_n} = \overline{F_n}F_n = nI_n$。于是 $F_n^{-1} = \frac{1}{n}\overline{F_n}$。

傅立叶矩阵

$$F_n = [\omega_n^{ij}]_{0\le i,j<n}(\omega_n = e^{\frac{2\pi i}{n}})$$

$$= \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \dots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \dots & \omega_n^{2(n-1)} \\ & & \dots & & \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)^2} \end{bmatrix}$$

$F_n\overline{F_n} = \overline{F_n}F_n = nI_n$。于是 $F_n^{-1} = \frac{1}{n}\overline{F_n}$。
$\frac{1}{\sqrt{n}}F_n$ 是酉矩阵（乘自己的共轭转置等于单位矩阵）。

DFT

DFT

给出多项式系数 $f(x) = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1}$，求其在 n 个 n 次单位根上的点值 $f(\omega_n), f(\omega_n^2), \ldots, f(\omega_n^{n-1})$。

DFT

给出多项式系数 $f(x)=a_0+a_1x+...+a_{n-1}x^{n-1}$，求其在 n 个 n 次单位根上的点值 $f(\omega_n), f(\omega_n^2), ..., f(\omega_n^{n-1})$。

写成矩阵：

$$A=\begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_{n-2} \\ a_{n-1} \end{bmatrix}, F=\begin{bmatrix} f(1) \\ f(\omega_n) \\ \dots \\ f(\omega_n^{n-2}) \\ f(\omega_n^{n-1}) \end{bmatrix}, F_n=\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \dots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \dots & \omega_n^{2(n-1)} \\ & & \dots & & \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)^2} \end{bmatrix}$$

DFT

给出多项式系数 $f(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1}$，求其在 n 个 n 次单位根上的点值 $f(\omega_n), f(\omega_n^2), ..., f(\omega_n^{n-1})$。
写成矩阵:

$$A = \begin{bmatrix} a_0 \\ a_1 \\ ... \\ a_{n-2} \\ a_{n-1} \end{bmatrix}, F = \begin{bmatrix} f(1) \\ f(\omega_n) \\ ... \\ f(\omega_n^{n-2}) \\ f(\omega_n^{n-1}) \end{bmatrix}, F_n = \begin{bmatrix} 1 & 1 & 1 & ... & 1 \\ 1 & \omega_n & \omega_n^2 & ... & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & ... & \omega_n^{2(n-1)} \\ & & ... & & \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & ... & \omega_n^{(n-1)^2} \end{bmatrix}$$

DFT:$F = F_n A$。

DFT

给出多项式系数 $f(x) = a_0 + a_1 x + ... + a_{n-1} x^{n-1}$，求其在 n 个 n 次单位根上的点值 $f(\omega_n), f(\omega_n^2), ..., f(\omega_n^{n-1})$。
写成矩阵：

$$A = \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_{n-2} \\ a_{n-1} \end{bmatrix}, F = \begin{bmatrix} f(1) \\ f(\omega_n) \\ \dots \\ f(\omega_n^{n-2}) \\ f(\omega_n^{n-1}) \end{bmatrix}, F_n = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \dots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \dots & \omega_n^{2(n-1)} \\ & & \dots & & \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)^2} \end{bmatrix}$$

DFT:$F = F_n A$。
IDFT:$A = F_n^{-1} F = \frac{1}{n} \overline{F_n} F$。

Contents

FFT

FFT

如果 n 是 2 的幂次，可以在 $O(n \log n)$ 时间内完成 DFT 和 IDFT。

FFT

如果 n 是 2 的幂次，可以在 $O(n\log n)$ 时间内完成 DFT 和 IDFT。

设 $A(x) = a_0 + a_1x + \ldots + a_{n-1}x^{n-1}$。

FFT

如果 n 是 2 的幂次，可以在 $O(n\log n)$ 时间内完成 DFT 和 IDFT。

设 $A(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1}$。

定义两个 $\frac{n}{2}$ 阶的多项式

FFT

如果 n 是 2 的幂次，可以在 $O(n\log n)$ 时间内完成 DFT 和 IDFT。

设 $A(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1}$。

定义两个 $\frac{n}{2}$ 阶的多项式

$$A_0(x) = a_0 + a_2 x + a_4 x^2 \ldots + a_{n-2}x^{\frac{n}{2}-1}$$

$$A_1(x) = a_1 + a_3 x + a_5 x^2 \ldots + a_{n-1}x^{\frac{n}{2}-1}$$

FFT

如果 n 是 2 的幂次，可以在 $O(n\log n)$ 时间内完成 DFT 和 IDFT。

设 $A(x) = a_0 + a_1x + ... + a_{n-1}x^{n-1}$。

定义两个 $\frac{n}{2}$ 阶的多项式

$$A_0(x) = a_0 + a_2x + a_4x^2 ... + a_{n-2}x^{\frac{n}{2}-1}$$

$$A_1(x) = a_1 + a_3x + a_5x^2 ... + a_{n-1}x^{\frac{n}{2}-1}$$

则 $A(x) = A_0(x^2) + xA_1(x^2)$。

FFT

FFT

设 $y_0 = DFT(A_0(x)), y_1 = DFT(A_1(x))$,

FFT

设 $y_0 = DFT(A_0(x)), y_1 = DFT(A_1(x))$，即

$$y_0[k] = A_0(\omega_{\frac{n}{2}}^k)$$

$$y_1[k] = A_1(\omega_{\frac{n}{2}}^k)$$

FFT

设 $y_0 = DFT(A_0(x)), y_1 = DFT(A_1(x))$，即

$$y_0[k] = A_0(\omega_{\frac{n}{2}}^{k})$$

$$y_1[k] = A_1(\omega_{\frac{n}{2}}^{k})$$

由 $\omega_{\frac{n}{2}}^{k} = \omega_n^{2k} = (\omega_n^k)^2$，

FFT

设 $y_0 = DFT(A_0(x)), y_1 = DFT(A_1(x))$，即

$$y_0[k] = A_0(\omega_{\frac{n}{2}}^k)$$

$$y_1[k] = A_1(\omega_{\frac{n}{2}}^k)$$

由 $\omega_{\frac{n}{2}}^k = \omega_n^{2k} = (\omega_n^k)^2$，有

$$y_0[k] = A_0((\omega_n^k)^2)$$

$$y_1[k] = A_1((\omega_n^k)^2)$$

FFT

$$y_0[k] = A_0((\omega_n^k)^2)$$
$$y_1[k] = A_1((\omega_n^k)^2)$$

FFT

$$y_0[k] = A_0((\omega_n^k)^2)$$
$$y_1[k] = A_1((\omega_n^k)^2)$$

计算 $y = DFT(A(x))$：

FFT

$$y_0[k] = A_0((\omega_n^k)^2)$$
$$y_1[k] = A_1((\omega_n^k)^2)$$

计算 $y = DFT(A(x))$：

$$y[k] = A(\omega_n^k) = A_0((\omega_n^k)^2) + \omega_n^k A_1((\omega_n^k)^2) = y_0[k] + \omega_n^k y_1[k]$$

$$y[k+\frac{n}{2}] = A(\omega_n^{k+\frac{n}{2}}) = A_0((\omega_n^{k+\frac{n}{2}})^2) + \omega_n^{k+\frac{n}{2}} A_1((\omega_n^{k+\frac{n}{2}})^2) = y_0[k] - \omega_n^k y_1[k]$$

FFT

$$y_0[k] = A_0((\omega_n^k)^2)$$
$$y_1[k] = A_1((\omega_n^k)^2)$$

计算 $y = DFT(A(x))$：

$$y[k] = A(\omega_n^k) = A_0((\omega_n^k)^2) + \omega_n^k A_1((\omega_n^k)^2) = y_0[k] + \omega_n^k y_1[k]$$

$$y[k+\frac{n}{2}] = A(\omega_n^{k+\frac{n}{2}}) = A_0((\omega_n^{k+\frac{n}{2}})^2)+\omega_n^{k+\frac{n}{2}} A_1((\omega_n^{k+\frac{n}{2}})^2) = y_0[k]-\omega_n^k y_1[k]$$

$T(n) = 2T(\frac{n}{2}) + O(n)$, $T(n) = O(n \log n)$。

NTT

NTT

就是把复数域上的单位根换成模域下 $g^{\frac{p-1}{n}}$（n 是 2 的幂次，g 是原根）。

NTT

就是把复数域上的单位根换成模域下 $g^{\frac{p-1}{n}}$（n 是 2 的幂次，g 是原根）。

显然它也满足单位根的性质：$g^n \equiv 1$，g^i 互不相同（$i \in [0, n)$）。

NTT

就是把复数域上的单位根换成模域下 $g^{\frac{p-1}{n}}$（n 是 2 的幂次，g 是原根）。

显然它也满足单位根的性质：$g^n \equiv 1$，g^i 互不相同（$i \in [0, n)$）。

模数 p 需要满足 2 的幂次整除 $p-1$。

NTT

就是把复数域上的单位根换成模域下 $g^{\frac{p-1}{n}}$（n 是 2 的幂次，g 是原根）。

显然它也满足单位根的性质：$g^n \equiv 1$，g^i 互不相同（$i \in [0, n)$）。

模数 p 需要满足 2 的幂次整除 $p-1$。

常见 NTT 模数：

- $998244353 = 7 * 17 * 2^{23} + 1$
- $1004535809 = 479 * 2^{21} + 1$
- $1998585857 = 953 * 2^{21} + 1$

模任意素数 FFT

模任意素数 FFT

方法大致有 3 种：

- 取三个 NTT 模数，CRT 合并
- 权值分两段做 FFT
- MTT

模任意素数 FFT-三模数合并

模任意素数 FFT-三模数合并

不取模的话，FFT 权值是 $O(P^2 n)$ 的。

模任意素数 FFT-三模数合并

不取模的话，FFT 权值是 $O(P^2 n)$ 的。
于是只要取 $O(\log(P^2 n))$ 个 NTT 模数，分别计算再用 CRT 合并起来即可。

模任意素数 FFT-三模数合并

不取模的话，FFT 权值是 $O(P^2 n)$ 的。

于是只要取 $O(\log(P^2 n))$ 个 NTT 模数，分别计算再用 CRT 合并起来即可。

比如 P 是 10^9 级别的数的时候，取 3 个 NTT 模数合并就行了。

模任意素数 FFT-权值分两段

模任意素数 FFT-权值分两段

令 $S = \sqrt{P} \simeq O(10^{4.5})$，把每个数表示成 $AS + B$。

模任意素数 FFT-权值分两段

令 $S=\sqrt{P}\simeq O(10^{4.5})$，把每个数表示成 $AS+B$。

$(A_1S+B_1)(A_2S+B_2)=(A_1A_2)S^2+(A_1B_2+A_2B_1)S+(B_1B_2)$

模任意素数 FFT-权值分两段

令 $S=\sqrt{P}\simeq O(10^{4.5})$，把每个数表示成 $AS+B$。
$(A_1S+B_1)(A_2S+B_2)=(A_1A_2)S^2+(A_1B_2+A_2B_1)S+(B_1B_2)$
每次 FFT 权值上限 $O(P^2n)\simeq O(10^{14})$，接近 double 的精度范围。

模任意素数 FFT-权值分两段

令 $S=\sqrt{P}\simeq O(10^{4.5})$，把每个数表示成 $AS+B$。

$(A_1S+B_1)(A_2S+B_2)=(A_1A_2)S^2+(A_1B_2+A_2B_1)S+(B_1B_2)$

每次 FFT 权值上限 $O(P^2n)\simeq O(10^{14})$，接近 double 的精度范围。

需要 4 次 DFT，3 次 IDFT。范围较大的话最好用 long double 以保证精度。

模任意素数 FFT-MTT

模任意素数 FFT-MTT

在权值分两段的基础上，用 MTT 的技巧将 DFT/IDFT 两两合并以减少次数。

模任意素数 FFT-MTT

在权值分两段的基础上，用 MTT 的技巧将 DFT/IDFT 两两合并以减少次数。

设 $P(x) = A(x) + B(x)i, Q(x) = A(x) - B(x)i,$

模任意素数 FFT-MTT

在权值分两段的基础上，用 MTT 的技巧将 DFT/IDFT 两两合并以减少次数。

设 $P(x) = A(x) + B(x)i, Q(x) = A(x) - B(x)i$，则

$$Q(\omega_n^k) = \overline{P(\omega_n^{-k})}$$

模任意素数 FFT-MTT

在权值分两段的基础上，用 MTT 的技巧将 DFT/IDFT 两两合并以减少次数。

设 $P(x) = A(x) + B(x)i, Q(x) = A(x) - B(x)i$，则

$$Q(\omega_n^k) = \overline{P(\omega_n^{-k})}$$

$$A(x) = \frac{P(x) + Q(x)}{2}, B(x) = \frac{Q(x) - P(x)}{2}i$$

模任意素数 FFT-MTT

在权值分两段的基础上，用 MTT 的技巧将 DFT/IDFT 两两合并以减少次数。

设 $P(x) = A(x) + B(x)i, Q(x) = A(x) - B(x)i$，则

$$Q(\omega_n^k) = \overline{P(\omega_n^{-k})}$$

$$A(x) = \frac{P(x) + Q(x)}{2}, B(x) = \frac{Q(x) - P(x)}{2}i$$

由 IDFT 完都是实多项式，可以把两个多项式的点值放在实部虚部一起 IDFT，然后取实部虚部即可。

模任意素数 FFT-MTT

在权值分两段的基础上，用 MTT 的技巧将 DFT/IDFT 两两合并以减少次数。

设 $P(x) = A(x) + B(x)i, Q(x) = A(x) - B(x)i$，则

$$Q(\omega_n^k) = \overline{P(\omega_n^{-k})}$$

$$A(x) = \frac{P(x) + Q(x)}{2}, B(x) = \frac{Q(x) - P(x)}{2}i$$

由 IDFT 完都是实多项式，可以把两个多项式的点值放在实部虚部一起 IDFT，然后取实部虚部即可。

这样总共需要 4 次 DFT/IDFT。

卷积

卷积

给出两个长为 n 的数组 A, B，求数组 C:

$$A_iB_j \to C_{i+j}$$

卷积

给出两个长为 n 的数组 A, B，求数组 C:

$$A_i B_j \to C_{i+j}$$

只要把 A, B 都看成长为 2 的幂次的多项式，DFT 求出单位根的点值，乘起来得到 C 的点值，IDFT 插回多项式即可。

卷积

给出两个长为 n 的数组 A, B，求数组 C:

$$A_i B_j \to C_{i+j}$$

只要把 A, B 都看成长为 2 的幂次的多项式，DFT 求出单位根的点值，乘起来得到 C 的点值，IDFT 插回多项式即可。

复杂度 $O(n \log n)$。

循环卷积

循环卷积

直接 DFT，点乘，IDFT 得出的是循环卷积的结果：

$$A_i B_j \to C_{(i+j) \bmod n}$$

循环卷积

直接 DFT，点乘，IDFT 得出的是循环卷积的结果：

$$A_i B_j \to C_{(i+j) \bmod n}$$

如果要求长度为 n（非 2 的幂次）的 DFT，n 不大可以直接 $O(n^2)$ 暴力，n 大的话可以使用 Bluestein 算法。

Bluestein's Algorithm

Bluestein's Algorithm

$$A(\omega_n^j) = \sum_{i=0}^{n-1} a_i \omega_n^{ij}$$

Bluestein's Algorithm

$$
\begin{aligned}
A(\omega_n^j) &= \sum_{i=0}^{n-1} a_i \omega_n^{ij} \\
&= \sum_{i=0}^{n-1} a_i \omega_{2n}^{i^2+j^2-(i-j)^2}
\end{aligned}
$$

Bluestein's Algorithm

$$
\begin{aligned}
A(\omega_n^j) &= \sum_{i=0}^{n-1} a_i \omega_n^{ij} \\
&= \sum_{i=0}^{n-1} a_i \omega_{2n}^{i^2+j^2-(i-j)^2} \\
&= \omega_{2n}^{j^2} \sum_{i=0}^{n-1} (a_i \omega_{2n}^{i^2}) \times (\omega_{2n}^{-(i-j)^2})
\end{aligned}
$$

Bluestein's Algorithm

$$
\begin{aligned}
A(\omega_n^j) &= \sum_{i=0}^{n-1} a_i \omega_n^{ij} \\
&= \sum_{i=0}^{n-1} a_i \omega_{2n}^{i^2+j^2-(i-j)^2} \\
&= \omega_{2n}^{j^2} \sum_{i=0}^{n-1} (a_i \omega_{2n}^{i^2}) \times (\omega_{2n}^{-(i-j)^2})
\end{aligned}
$$

这是个卷积形式，一遍 FFT 即可。

高维 DFT

高维 DFT

设 $n = n_0 n_1 \dots n_{m-1}$，考虑 m 元多项式

$$f(x_0, \dots, x_{m-1}) = \sum_{i_0=0}^{n_0-1} \dots \sum_{i_{m-1}=0}^{n_{m-1}-1} f_{\overline{i_0 i_1 \dots i_{m-1}}} x_0^{i_0} \dots x_{m-1}^{i_{m-1}}$$

高维 DFT

设 $n = n_0 n_1 \ldots n_{m-1}$，考虑 m 元多项式

$$f(x_0, \ldots, x_{m-1}) = \sum_{i_0=0}^{n_0-1} \cdots \sum_{i_{m-1}=0}^{n_{m-1}-1} f_{\overline{i_0 i_1 \ldots i_{m-1}}} x_0^{i_0} \ldots x_{m-1}^{i_{m-1}}$$

它的 DFT 是一个长为 n 的数组，其中第 $\overline{i_0 i_1 \ldots i_{m-1}}$ 个位置存放的是 $f(\omega_{n_0}^{i_0}, \omega_{n_1}^{i_1}, \ldots, \omega_{n_{m-1}}^{i_{m-1}})$。

高维 DFT

设 $n = n_0 n_1 \dots n_{m-1}$，考虑 m 元多项式

$$f(x_0, \dots, x_{m-1}) = \sum_{i_0=0}^{n_0-1} \dots \sum_{i_{m-1}=0}^{n_{m-1}-1} f_{\overline{i_0 i_1 \dots i_{m-1}}} x_0^{i_0} \dots x_{m-1}^{i_{m-1}}$$

它的 DFT 是一个长为 n 的数组，其中第 $\overline{i_0 i_1 \dots i_{m-1}}$ 个位置存放的是 $f(\omega_{n_0}^{i_0}, \omega_{n_1}^{i_1}, \dots, \omega_{n_{m-1}}^{i_{m-1}})$。

设 n_0 个长为 $\frac{n}{n_0}$ 的子多项式 $f_0, f_1, \dots, f_{n_0-1}$ 分别表示去掉 x_0，i_0 取 $0 \sim n_0 - 1$ 时的 $m-1$ 元多项式。

高维 DFT

设 $n = n_0 n_1 \dots n_{m-1}$，考虑 m 元多项式

$$f(x_0, \dots, x_{m-1}) = \sum_{i_0=0}^{n_0-1} \dots \sum_{i_{m-1}=0}^{n_{m-1}-1} f_{\overline{i_0 i_1 \dots i_{m-1}}} x_0^{i_0} \dots x_{m-1}^{i_{m-1}}$$

它的 DFT 是一个长为 n 的数组，其中第 $\overline{i_0 i_1 \dots i_{m-1}}$ 个位置存放的是 $f(\omega_{n_0}^{i_0}, \omega_{n_1}^{i_1}, \dots, \omega_{n_{m-1}}^{i_{m-1}})$。

设 n_0 个长为 $\frac{n}{n_0}$ 的子多项式 $f_0, f_1, \dots, f_{n_0-1}$ 分别表示去掉 x_0，i_0 取 $0 \sim n_0 - 1$ 时的 $m - 1$ 元多项式。设 $F_0, F_1, \dots, F_{n_0-1}$ 分别为它们的 DFT，则有

高维 DFT

高维 DFT

$$
\begin{aligned}
F = \{ & (\omega_{n_0}^0)^0 F_0 + (\omega_{n_0}^0)^1 F_1 + \ldots + (\omega_{n_0}^0)^{n_0-1} F_{n_0-1}, \\
& (\omega_{n_0}^1)^0 F_0 + (\omega_{n_1}^0)^1 F_1 + \ldots + (\omega_{n_0}^1)^{n_0-1} F_{n_0-1}, \\
& \ldots \\
& (\omega_{n_0}^{n_0-1})^0 F_0 + (\omega_{n_0}^{n_0-1})^1 F_1 + \ldots + (\omega_{n_0}^{n_0-1})^{n_0-1} F_{n_0-1} \}
\end{aligned}
$$

高维 DFT

$$
\begin{aligned}
F = \{&(\omega_{n_0}^{0})^{0} F_0 + (\omega_{n_0}^{0})^{1} F_1 + \ldots + (\omega_{n_0}^{0})^{n_0-1} F_{n_0-1},\\
&(\omega_{n_0}^{1})^{0} F_0 + (\omega_{n_1}^{0})^{1} F_1 + \ldots + (\omega_{n_0}^{1})^{n_0-1} F_{n_0-1},\\
&\ldots\\
&(\omega_{n_0}^{n_0-1})^{0} F_0 + (\omega_{n_0}^{n_0-1})^{1} F_1 + \ldots + (\omega_{n_0}^{n_0-1})^{n_0-1} F_{n_0-1}\}
\end{aligned}
$$

逗号表示连接。

高维 DFT

$$
\begin{aligned}
F = \{ & (\omega_{n_0}^0)^0 F_0 + (\omega_{n_0}^0)^1 F_1 + \ldots + (\omega_{n_0}^0)^{n_0-1} F_{n_0-1}, \\
& (\omega_{n_0}^1)^0 F_0 + (\omega_{n_1}^0)^1 F_1 + \ldots + (\omega_{n_0}^1)^{n_0-1} F_{n_0-1}, \\
& \ldots \\
& (\omega_{n_0}^{n_0-1})^0 F_0 + (\omega_{n_0}^{n_0-1})^1 F_1 + \ldots + (\omega_{n_0}^{n_0-1})^{n_0-1} F_{n_0-1} \}
\end{aligned}
$$

逗号表示连接。

每一维如果暴力 DFT 的话，复杂度是 $T(n) = T(\frac{n}{n_0}) + O(n_0^2 \frac{n}{n_0})$，$n_i$ 都很小的时候可以认为 $T(n) = O(n \log n)$。

高维 DFT

$$
\begin{aligned}
F = \{&(\omega_{n_0}^{0})^{0}F_0 + (\omega_{n_0}^{0})^{1}F_1 + \ldots + (\omega_{n_0}^{0})^{n_0-1}F_{n_0-1},\\
&(\omega_{n_0}^{1})^{0}F_0 + (\omega_{n_1}^{0})^{1}F_1 + \ldots + (\omega_{n_0}^{1})^{n_0-1}F_{n_0-1},\\
&\ldots\\
&(\omega_{n_0}^{n_0-1})^{0}F_0 + (\omega_{n_0}^{n_0-1})^{1}F_1 + \ldots + (\omega_{n_0}^{n_0-1})^{n_0-1}F_{n_0-1}\}
\end{aligned}
$$

逗号表示连接。

每一维如果暴力 DFT 的话，复杂度是 $T(n) = T(\frac{n}{n_0}) + O(n_0^2\frac{n}{n_0})$，$n_i$ 都很小的时候可以认为 $T(n) = O(n\log n)$。

如果某些维长度是 2 的幂次也可以在这些维上 FFT。

Contents

Contents

多项式求逆

多项式求逆

给出 n 阶多项式 $A(x)$，求一个 n 阶多项式 $B(x)$ 使得

$$A(x)B(x) \equiv 1 \pmod{x^n}$$

多项式求逆

多项式求逆

当 $n = 1$ 时答案就是逆元。

多项式求逆

当 $n = 1$ 时答案就是逆元。
假设已经求出了 $\lceil \frac{n}{2} \rceil$ 时的答案。

多项式求逆

当 $n = 1$ 时答案就是逆元。

假设已经求出了 $\lceil \frac{n}{2} \rceil$ 时的答案。即 $\lceil \frac{n}{2} \rceil$ 阶多项式 $B_0(x)$ 使得 $A(x)B_0(x) \equiv 1 \pmod{x^{\lceil \frac{n}{2} \rceil}}$。

多项式求逆

当 $n=1$ 时答案就是逆元。

假设已经求出了 $\lceil \frac{n}{2} \rceil$ 时的答案。即 $\lceil \frac{n}{2} \rceil$ 阶多项式 $B_0(x)$ 使得 $A(x)B_0(x) \equiv 1 \pmod{x^{\lceil \frac{n}{2} \rceil}}$。

移项平方，

$$A^2(x)B_0^2(x) - 2A(x)B_0(x) + 1 \equiv 0 \pmod{x^n}$$

多项式求逆

当 $n=1$ 时答案就是逆元。

假设已经求出了 $\lceil \frac{n}{2} \rceil$ 时的答案。即 $\lceil \frac{n}{2} \rceil$ 阶多项式 $B_0(x)$ 使得 $A(x)B_0(x) \equiv 1 \pmod{x^{\lceil \frac{n}{2} \rceil}}$。

移项平方，

$$A^2(x)B_0^2(x) - 2A(x)B_0(x) + 1 \equiv 0 \pmod{x^n}$$

两边同乘 $B(x)$，

$$A(x)B_0^2(x) - 2B_0(x) + B(x) \equiv 0 \pmod{x^n}$$

多项式求逆

当 $n = 1$ 时答案就是逆元。

假设已经求出了 $\lceil \frac{n}{2} \rceil$ 时的答案。即 $\lceil \frac{n}{2} \rceil$ 阶多项式 $B_0(x)$ 使得 $A(x)B_0(x) \equiv 1 \pmod{x^{\lceil \frac{n}{2} \rceil}}$。

移项平方，

$$A^2(x)B_0^2(x) - 2A(x)B_0(x) + 1 \equiv 0 \pmod{x^n}$$

两边同乘 $B(x)$，

$$A(x)B_0^2(x) - 2B_0(x) + B(x) \equiv 0 \pmod{x^n}$$

$T(n) = T(n/2) + O(n \log n)$, $T(n) = O(n \log n)$。

Contents

多项式取模

多项式取模

给出 n 阶多项式 $A(x)$ 和 m 阶多项式 $B(x)$，求一个 $n-m$ 阶多项式 $D(x)$ 和一个 $m-1$ 阶多项式 $R(x)$，使得

$$A(x) = B(x)D(x) + R(x)$$

多项式取模

多项式取模

当 $n < m$ 的时候令 $D(x) = 0, R(x) = A(x)$ 就行了。

多项式取模

当 $n < m$ 的时候令 $D(x) = 0, R(x) = A(x)$ 就行了。
设 $F^R(x) = x^{n-1}F(\frac{1}{x})$，即把 $F(x)$ 的系数翻转的结果。

多项式取模

当 $n < m$ 的时候令 $D(x) = 0, R(x) = A(x)$ 就行了。
设 $F^R(x) = x^{n-1}F(\frac{1}{x})$，即把 $F(x)$ 的系数翻转的结果。

$$A\left(\frac{1}{x}\right) = B\left(\frac{1}{x}\right) D\left(\frac{1}{x}\right) + R\left(\frac{1}{x}\right)$$

多项式取模

当 $n < m$ 的时候令 $D(x) = 0, R(x) = A(x)$ 就行了。
设 $F^R(x) = x^{n-1}F(\frac{1}{x})$，即把 $F(x)$ 的系数翻转的结果。

$$A\left(\frac{1}{x}\right) = B\left(\frac{1}{x}\right) D\left(\frac{1}{x}\right) + R\left(\frac{1}{x}\right)$$

两边同乘 x^{n-1}，有

$$A^R(x) = B^R(x)D^R(x) + R^R(x) * x^{n-m+1}$$

多项式取模

当 $n < m$ 的时候令 $D(x) = 0, R(x) = A(x)$ 就行了。
设 $F^R(x) = x^{n-1}F(\frac{1}{x})$，即把 $F(x)$ 的系数翻转的结果。

$$A\left(\frac{1}{x}\right) = B\left(\frac{1}{x}\right) D\left(\frac{1}{x}\right) + R\left(\frac{1}{x}\right)$$

两边同乘 x^{n-1}，有

$$A^R(x) = B^R(x)D^R(x) + R^R(x) * x^{n-m+1}$$

发现 D 的阶数是 $n-m+1$，那么在模 x^{n-m+1} 意义下，$D^R(x) = \frac{A^R(x)}{B^R(x)}$。

多项式取模

当 $n < m$ 的时候令 $D(x) = 0, R(x) = A(x)$ 就行了。
设 $F^R(x) = x^{n-1}F(\frac{1}{x})$，即把 $F(x)$ 的系数翻转的结果。

$$A\left(\frac{1}{x}\right) = B\left(\frac{1}{x}\right) D\left(\frac{1}{x}\right) + R\left(\frac{1}{x}\right)$$

两边同乘 x^{n-1}，有

$$A^R(x) = B^R(x)D^R(x) + R^R(x) * x^{n-m+1}$$

发现 D 的阶数是 $n-m+1$，那么在模 x^{n-m+1} 意义下，$D^R(x) = \frac{A^R(x)}{B^R(x)}$。
对 $B^R(x)$ 多项式求逆，卷上 $A^R(x)$，求出 $D^R(x)$，代进原式里得出 $R(x)$。$O(n\log n)$。

常系数线性递推

常系数线性递推

给出常系数线性递推式 $f_i = \sum_{j=1}^{k} c_j f_{i-j}$ 以及 $f_1 \ldots f_k$，求 f_n。

常系数线性递推

给出常系数线性递推式 $f_i = \sum_{j=1}^{k} c_j f_{i-j}$ 以及 $f_1 \dots f_k$，求 f_n。

$$\begin{bmatrix} f_2 \\ f_3 \\ \dots \\ f_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & & \\ 0 & 0 & 1 & \\ & & \dots & \\ c_k & c_{k-1} & \dots & c_1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_k \end{bmatrix}$$

常系数线性递推

给出常系数线性递推式 $f_i = \sum_{j=1}^{k} c_j f_{i-j}$ 以及 $f_1 \ldots f_k$，求 f_n。

$$\begin{bmatrix} f_2 \\ f_3 \\ \ldots \\ f_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & & \\ 0 & 0 & 1 & \\ & & \ldots & \\ c_k & c_{k-1} & \ldots & c_1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \ldots \\ f_k \end{bmatrix}$$

转移矩阵称为 Frobenius 矩阵，特征多项式为 $x^k - \sum_{i=1}^{k} c_i x^{k-i}$。

常系数线性递推

给出常系数线性递推式 $f_i=\sum_{j=1}^{k} c_j f_{i-j}$ 以及 $f_1 \dots f_k$，求 f_n。

$$\begin{bmatrix} f_2 \\ f_3 \\ \dots \\ f_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & & \\ 0 & 0 & 1 & \\ & & \dots & \\ c_k & c_{k-1} & \dots & c_1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_k \end{bmatrix}$$

转移矩阵称为 Frobenius 矩阵，特征多项式为 $x^k-\sum_{i=1}^{k} c_i x^{k-i}$。要求 $A^{n-1}V[1]$。

常系数线性递推

常系数线性递推

Cayley-Hamilton 定理：$F(A) = 0$，$F(x)$ 是 A 的特征多项式。

常系数线性递推

Cayley-Hamilton 定理：$F(A) = 0$，$F(x)$ 是 A 的特征多项式。
于是若 $x^{n-1} = F(x)D(x) + R(x)$

常系数线性递推

Cayley-Hamilton 定理：$F(A)=0$，$F(x)$ 是 A 的特征多项式。于是若 $x^{n-1}=F(x)D(x)+R(x)$，则

$$A^{n-1}=F(A)D(A)+R(A)=R(A)=\sum_{i=0}^{k-1}r_iA^i$$

常系数线性递推

Cayley-Hamilton 定理：$F(A)=0$，$F(x)$ 是 A 的特征多项式。于是若 $x^{n-1}=F(x)D(x)+R(x)$，则

$$A^{n-1}=F(A)D(A)+R(A)=R(A)=\sum_{i=0}^{k-1} r_i A^i$$

$$A^{n-1}V[1]=\sum_{i=0}^{k-1} r_i A^i V[1]=\sum_{i=0}^{k-1} r_i f_{i+1}$$

常系数线性递推

Cayley-Hamilton 定理：$F(A)=0$，$F(x)$ 是 A 的特征多项式。
于是若 $x^{n-1}=F(x)D(x)+R(x)$，则

$$A^{n-1}=F(A)D(A)+R(A)=R(A)=\sum_{i=0}^{k-1} r_i A^i$$

$$A^{n-1}V[1]=\sum_{i=0}^{k-1} r_i A^i V[1]=\sum_{i=0}^{k-1} r_i f_{i+1}$$

倍增 + 多项式取模，$O(k\log k\log n)$。

Contents

多点求值

多点求值

给出一个 m 阶多项式 $A(x)$ 和 n 个点 $x_1, x_2 \ldots x_n$，求这些点的点值。

多点求值

多点求值

令 $mid = n/2$。设多项式

多点求值

令 $mid = n/2$。设多项式

$$P_0(x) = \prod_{i=1}^{mid} (x - x_i)\,, P_1(x) = \prod_{i=mid+1}^{n} (x - x_i)$$

多点求值

令 $mid = n/2$。设多项式

$$P_0(x) = \prod_{i=1}^{mid} (x - x_i)\,, P_1(x) = \prod_{i=mid+1}^{n} (x - x_i)$$

$$A_0(x) = A(x) \bmod P_0(x), A_1(x) = A(x) \bmod P_1(x)$$

多点求值

令 $mid = n/2$。设多项式

$$P_0(x) = \prod_{i=1}^{mid} (x - x_i)\,, P_1(x) = \prod_{i=mid+1}^{n} (x - x_i)$$

$$A_0(x) = A(x) \bmod P_0(x), A_1(x) = A(x) \bmod P_1(x)$$

即

$$A(x) = P_0(x)D_0(x) + A_0(x) = P_1(x)D_0(x) + A_1(x)$$

多点求值

令 $mid = n/2$。设多项式

$$P_0(x) = \prod_{i=1}^{mid} (x - x_i)\,, P_1(x) = \prod_{i=mid+1}^{n} (x - x_i)$$

$$A_0(x) = A(x) \bmod P_0(x), A_1(x) = A(x) \bmod P_1(x)$$

即

$$A(x) = P_0(x)D_0(x) + A_0(x) = P_1(x)D_0(x) + A_1(x)$$

可以发现 $A(x)$ 在前后半的点处的点值分别等于 $A_0(x)$ 和 $A_1(x)$ 的点值。那么递归下去做就行了。

多点求值

令 $mid = n/2$。设多项式

$$P_0(x) = \prod_{i=1}^{mid} (x - x_i), P_1(x) = \prod_{i=mid+1}^{n} (x - x_i)$$

$$A_0(x) = A(x) \bmod P_0(x), A_1(x) = A(x) \bmod P_1(x)$$

即

$$A(x) = P_0(x)D_0(x) + A_0(x) = P_1(x)D_0(x) + A_1(x)$$

可以发现 $A(x)$ 在前后半的点处的点值分别等于 $A_0(x)$ 和 $A_1(x)$ 的点值。那么递归下去做就行了。$T(n) = O(n \log^2 n)$。

Contents

快速插值

快速插值

给出 n 个不同的点处的点值 (x_i, y_i)，求唯一的 n 阶多项式 $F(x)$。

快速插值

给出 n 个不同的点处的点值 (x_i, y_i)，求唯一的 n 阶多项式 $F(x)$。

拉格朗日插值：

快速插值

给出 n 个不同的点处的点值 (x_i, y_i)，求唯一的 n 阶多项式 $F(x)$。

拉格朗日插值：

$$F(x) = \sum_{i=1}^{n} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

快速插值

快速插值

$$F(x) = \sum_{i=1}^{n} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

快速插值

$$F(x)=\sum_{i=1}^{n} y_i \prod_{j\neq i} \frac{x-x_j}{x_i-x_j}$$

设 $G(x)=\prod_{i=1}^{n}(x-x_i)$，则有

快速插值

$$F(x)=\sum_{i=1}^{n} y_i \prod_{j\neq i} \frac{x-x_j}{x_i-x_j}$$

设 $G(x)=\prod_{i=1}^{n}(x-x_i)$，则有
$G'(x)=\sum_{i=1}^{n}\prod_{j\neq i}(x-x_j)$

快速插值

$$F(x) = \sum_{i=1}^{n} y_i \prod_{j\neq i} \frac{x - x_j}{x_i - x_j}$$

设 $G(x) = \prod_{i=1}^{n}(x - x_i)$，则有
$G'(x) = \sum_{i=1}^{n} \prod_{j\neq i}(x - x_j)$
$G'(x_i) = \prod_{j\neq i}(x_i - x_j)$

快速插值

$$F(x)=\sum_{i=1}^{n} y_i \prod_{j\neq i} \frac{x-x_j}{x_i-x_j}$$

设 $G(x)=\prod_{i=1}^{n}(x-x_i)$，则有
$G'(x)=\sum_{i=1}^{n}\prod_{j\neq i}(x-x_j)$
$G'(x_i)=\prod_{j\neq i}(x_i-x_j)$ ，多点求值 $O(n\log^2 n)$ 求出。

快速插值

$$F(x) = \sum_{i=1}^{n} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

设 $G(x) = \prod_{i=1}^{n}(x - x_i)$，则有
$G'(x) = \sum_{i=1}^{n} \prod_{j \neq i}(x - x_j)$
$G'(x_i) = \prod_{j \neq i}(x_i - x_j)$ ，多点求值 $O(n \log^2 n)$ 求出。

$$F(x) = \sum_{i=1}^{n} \frac{y_i}{G'(x_i)} \prod_{j \neq i}(x - x_j)$$

快速插值

$$F(x)=\sum_{i=1}^{n} y_i \prod_{j\neq i} \frac{x-x_j}{x_i-x_j}$$

设 $G(x)=\prod_{i=1}^{n}(x-x_i)$，则有
$G'(x)=\sum_{i=1}^{n}\prod_{j\neq i}(x-x_j)$
$G'(x_i)=\prod_{j\neq i}(x_i-x_j)$ ，多点求值 $O(n\log^2 n)$ 求出。

$$F(x)=\sum_{i=1}^{n}\frac{y_i}{G'(x_i)}\prod_{j\neq i}(x-x_j)$$

分治，左边乘上右边的 $\prod(x-x_j)$，右边乘上左边的 $\prod(x-x_j)$，加起来。

快速插值

$$F(x)=\sum_{i=1}^{n} y_i \prod_{j\neq i} \frac{x-x_j}{x_i-x_j}$$

设 $G(x)=\prod_{i=1}^{n}(x-x_i)$，则有
$G'(x)=\sum_{i=1}^{n}\prod_{j\neq i}(x-x_j)$
$G'(x_i)=\prod_{j\neq i}(x_i-x_j)$ ，多点求值 $O(n\log^2 n)$ 求出。

$$F(x)=\sum_{i=1}^{n}\frac{y_i}{G'(x_i)}\prod_{j\neq i}(x-x_j)$$

分治，左边乘上右边的 $\prod(x-x_j)$，右边乘上左边的 $\prod(x-x_j)$，加起来。$O(n\log^2 n)$。