Fourier transform

郭晓旭

上海交通大学

2014 年 6 月 6 日

动机：多项式乘法

多项式乘法

多项式 $A(x) = \sum a_n x^n$ 和 $B(x) = \sum b_n x^n$，求 $A(x) \cdot B(x)$。

系数和点值表示法

对于 $N-1$ 次多项式 $f(x)$，可以有 2 种不同的表示方法：

系数表示法 $f(x)=\sum_{0\le n<N} c_n x^n$；

点值表示法 $(x_1,x_2,\dots,x_N),(y_1,y_2,\dots,y_N)$ 满足 $f(x_i)=y_i$，即 (x_i,y_i) 是曲线上 $y=f(x)$ 的点。

系数表示法和点值表示法可以互相转化。

系数表示法 $\rightarrow$ 点值表示法

$$y_i = \sum_{0 \le n < N} c_i x_i^n$$

点值表示法 $\rightarrow$ 系数表示法

$$f(x) = \sum_{0 \le i < N} y_i \frac{\prod_{j \ne i}(x - x_j)}{\prod_{j \ne i}(x_i - x_j)}$$

对于多项式 $A(x)$ 和 $B(x)$，假设 $\deg A + \deg B < N$。
如果有 A 和 B 在 $\{x_0, x_1, \ldots, x_{N-1}\}$ 处的点值表示，则 $(A \cdot B)$ 的点值表示可以通过

$$(A \cdot B)(x_i) = A(x_i) \cdot B(x_i)$$

在 $O(N)$ 时间内得到。
还原 $(A \cdot B)$ 为系数表示就实现了多项式乘法。

对于多项式 $A(x)$ 和 $B(x)$，假设 $\deg A + \deg B < N$。
如果有 A 和 B 在 $\{x_0, x_1, \ldots, x_{N-1}\}$ 处的点值表示，则 $(A \cdot B)$ 的点值表示可以通过

$$(A \cdot B)(x_i) = A(x_i) \cdot B(x_i)$$

在 $O(N)$ 时间内得到。
还原 $(A \cdot B)$ 为系数表示就实现了多项式乘法。
变换的时间复杂度 $O(N^2)$。

Discrete Fourier transform

考虑在 $1, \omega, \omega^2, \ldots, \omega^{N-1}$ 的点值表示（其中 ω 是 N 次单位复根），即

$$y_k = \sum_{i=0}^{N-1} c_i (\omega^k)^i$$

假设 $N = 2^K$，$(y_0, y_1, \ldots, y_{N-1})$ 可以快速求出。

Fast Fourier transform[1]

$$f(x) = \sum_{i=0}^{N/2-1} a_{2i}(x^2)^i + x \sum_{i=0}^{N/2-1} a_{2i+1}(x^2)^i$$

当 x 取遍所有 N 次单位复根时，x^2 取遍所有 $(N/2)$ 次单位复根。所以只需计算多项式

$$f_0(x) = \sum_{i=0}^{N/2-1} a_{2i}x^i$$

和

$$f_1(x) = \sum_{i=0}^{N/2-1} a_{2i+1}x^i$$

的 DFT。

[1] https://github.com/ftiasch/shoka/blob/master/source/fast-fourier-transform.cpp

Inverse discrete Fourier transform

$$
\begin{aligned}
y_k &= \sum_{i=0}^{N-1} c_i (\omega^k)^i \\
c_k &= \frac{1}{N} \sum_{i=0}^{N-1} y_i (\omega^{-k})^i \\
&= \frac{1}{N} \sum_{i=0}^{N-1} (\sum_{j=0}^{N-1} c_j (\omega^i)^j)(\omega^{-k})^i \\
&= \frac{1}{N} \sum_{j=0}^{N-1} c_j (\sum_{i=0}^{N-1} (\omega^{j-k})^i) = c_k
\end{aligned}
$$

当 $k \neq 0$ 时，

$$
1 + \omega^k + (\omega^k)^2 + \cdots + (\omega^k)^{N-1} = \frac{1-(\omega^k)^N}{1-\omega^k} = 0
$$

Number theoretic transform

$\{1, \omega, \omega^2, \omega^3, \ldots\}$ 是 2^K 阶的循环群。
对于质数 $P = 2^K \cdot Q + 1$ 的原根 g，

$$\{1, g, g^2, g^3, \ldots\}$$

是 $2^K \cdot Q$ 阶的循环群。即

$$\{1, g^Q, g^{2Q}, g^{3Q}, \ldots\}$$

是 2^K 阶循环群，用 g 替代 ω 即可。

Triple Sums[2]

N 个整数 $A_1, A_2, \ldots, A_N$，对于所有的 S，求满足：

- $A_i + A_j + A_k = S$
- $i < j < k$

的 (i, j, k) 数量。
$(N \le 40000, A_i \le 20000)$

[2]SPOJ TSUM

暂且忽略 $i < j < k$ 的要求，构造多项式

$$A(x) = \sum_{1 \le i \le N} x^{A_i},$$

则 $A^3(x)$ 中 x^S 项的系数就是所求结果。

容斥原理

$$
\begin{aligned}
(\sum x)^3 &= \sum x^3 + 3\sum x^2 y + 6\sum xyz \\
(\sum x^2)(\sum x) &= \sum x^3 + \sum x^2 y \\
(\sum x^3) &= \sum x^3
\end{aligned}
$$

即

$$
\sum xyz = \frac{(\sum x)^3 - 3(\sum x^2)(\sum x) + 2(\sum x^3)}{6}
$$

Super Rooks on Chessboard[3]

超级车可以攻击行、列、主对角线 3 个方向。
$R \times C$ 的棋盘上有 N 个超级车，问被攻击的格子总数。
$(R, C, N \leq 50000)$

[3] ACM ICPC World Finals 2013 Warmup

设 R, C, D 分别为行上、列上、对角线上有车的格子集合，结果即为 $|R \cup C \cup D|$。

设 R, C, D 分别为行上、列上、对角线上有车的格子集合，结果即为 $|R \cup C \cup D|$。

$$|R \cup C \cup D| = |R| + |C| + |D| - |R \cap C| - |C \cap D| - |D \cap R| + |R \cap C \cap D|$$

设

$$
\begin{aligned}
R &= \{r_1, r_2, \ldots, r_n\} \\
C &= \{c_1, c_2, \ldots, c_m\} \\
D &= \{d_1, d_2, \ldots, d_k\}
\end{aligned}
$$

即计算 $r_i - c_j = d_k$ 的 (i, j, k) 数量。

设

$$
\begin{aligned}
R &= \{r_1, r_2, \ldots, r_n\} \\
C &= \{c_1, c_2, \ldots, c_m\} \\
D &= \{d_1, d_2, \ldots, d_k\}
\end{aligned}
$$

即计算 $r_i - c_j = d_k$ 的 (i, j, k) 数量。
构造多项式

$$R(x) = \sum_{i=1}^{n} x^{r_i},$$

$$C(x) = \sum_{i=1}^{m} x^{-c_i}$$

，计算 $(R \cdot C)(x)$ 中 x^{d_k} 的系数。

3-idiots

给出 $A_1, A_2, \ldots, A_N$，问随机选择一个三元子集，选择的数字是三角形的三边长的概率。
$(N \leq 10^5, 1 \leq A_i \leq 10^5)$

1. 求满足 $A_i + A_j > A_k$ 的 (i, j, k) 数量，$O(N + M \log M)$
2. 求满足 $A_i + A_j > A_k$ 且 $A_i \le A_j \le A_k$ 的数量

Arithmetic Progressions[4]

给出 $A_1, A_2, \ldots, A_N$ 统计满足:

- $i < j < k$
- $A_i + A_k = 2A_j$

的 (i, j, k) 数量。
$(N \leq 30000, A_i \leq 10^5)$

[4] Codechef November Challenge 2012

分块，假设块的大小是 C，考虑第 b 个块，有两种情况：

- $\{i,j,k\}$ 至少 2 个数在第 b 块中，$O((\frac{N}{C})^2)$
- i 在前 $(b-1)$ 块，j 在第 b 块，k 在后 $(n-b)$ 块，$O(C+M\log M)$

Normal[5]

N 个点的树，点分治时等概率地随机选点，代价为当前连通块的顶点数量，求代价的期望值。
$(N \leq 30000)$

[5]"我们都爱 GYZ 杯"NOIp 模拟赛

由期望的线性性，考虑点对 (u, v)，如果当 u 被选为分治中心时 v 和 u 连通，对代价贡献 $+1$。
点 u 到点 v 路径上共有 $\rho(u, v) + 1$ 个点，每个点被选择的概率相等，结果为

$$\sum_{u,v} \frac{1}{\rho(u, v) + 1}$$

需要对所有 $0 \le k \le N - 1$，统计 $\rho(u, v) = k$ 的 (u, v) 数量 C_k。

由期望的线性性，考虑点对 (u,v)，如果当 u 被选为分治中心时 v 和 u 连通，对代价贡献 $+1$。
点 u 到点 v 路径上共有 $\rho(u,v)+1$ 个点，每个点被选择的概率相等，结果为

$$\sum_{u,v} \frac{1}{\rho(u,v)+1}$$

需要对所有 $0 \le k \le N-1$，统计 $\rho(u,v)=k$ 的 (u,v) 数量 C_k。
用（标准的）点分治，假设点 u 的深度是 d_u，构造多项式

$$D(x) = \sum_u x^{d_u}$$

对 C_k 的贡献即 $D^2(x)$ 中 x^k 的系数。
减去来自相同子树的点对。时间复杂度 $O(N\log^2 N)$。

Point Distance[6]

$N \times N$ 的点阵，(x, y) 位置有 $C_{x,y}$ 个点，考虑所有点对，把点对按照 Euclidean 距离从小到大输出。
$(N \leq 1024)$

[6] JAG 2013 Spring

Point Distance[6]

$N \times N$ 的点阵，(x, y) 位置有 $C_{x,y}$ 个点，考虑所有点对，把点对按照 Euclidean 距离从小到大输出。
$(N \leq 1024)$
二维卷积，计算

$$D_{x,y} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C_{i,j} C_{i+x,j+y}$$

作映射 $\phi(x, y) = x \cdot 2N + y$，则

$$D_{\phi(x,y)} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C_{\phi(i,j)} C_{\phi(i,j)+\phi(x,y)}$$

[6] JAG 2013 Spring

SumOfArrays[7]

数列 $\{A_1, A_2, \ldots, A_N\}, \{B_1, B_2, \ldots, B_N\}$
任意改变元素的顺序，使 $A_i + B_i$ 的众数重数最大。
$1 \leq N \leq 10^5, 0 \leq A_i, B_i < 10^5$，数据随机

[7] Single Round Match 603, Level 3

设 U_k, V_k 表示 k 在 A, B 中的出现次数，W_k 表示 k 在 $A_i + B_i$ 的出现次数。则

$$W_k = \sum_{i=-\infty}^{\infty} \min\{U_i, V_{k-i}\}$$

设 U_k, V_k 表示 k 在 A, B 中的出现次数，W_k 表示 k 在 $A_i + B_i$ 的出现次数。则

$$W_k = \sum_{i=-\infty}^{\infty} \min\{U_i, V_{k-i}\}$$

因为

$$\min\{x, y\} = \sum_{n=1}^{\infty} [x \geq n][y \geq n]$$

，所以

$$\begin{aligned} W_k &= \sum_{i=-\infty}^{\infty} \sum_{j=1}^{\infty} [U_i \geq j][V_{k-i} \geq j] \\ &= \sum_{j=1}^{\infty} \sum_{i=-\infty}^{\infty} [U_i \geq j][V_{k-i} \geq j] \end{aligned}$$

$$W_k = \sum_{j=1}^{\infty} \sum_{i=-\infty}^{\infty} [U_i \geq j][V_{k-i} \geq j]$$

设 $M = O(\frac{\log N}{\log \log N})$，把 N 个球随机丢进 N 个盒子，盒子中最大球数很大概率是 M。

因此对于超过的盒子直接平方枚举即可。

时间复杂度约是 $O(\frac{\log N}{\log \log N} N \log N)$

Pattern matching

模板串 P 和文本串 T 都带有? 号，可以匹配任意一个字符，求 P 在 T 中所有的出现位置。

Pattern matching

模板串 P 和文本串 T 都带有? 号，可以匹配任意一个字符，求 P 在 T 中所有的出现位置。
假设 $|P| = M$，且没有通配符，计算

$$X_k = \sum_{i=0}^{M-1} (T_{k+i} - P_i)^2 = \sum_{i=0}^{M-1} T_{k+i}^2 - 2\sum_{i=0}^{M-1} T_{k+i}P_i + \sum_{i=0}^{M-1} P_i^2$$

则

$$X_k = 0 \iff \forall_{0 \le i < M} T_{k+i} = P_i$$

Pattern matching

模板串 P 和文本串 T 都带有? 号，可以匹配任意一个字符，求 P 在 T 中所有的出现位置。
假设 $|P| = M$，且没有通配符，计算

$$X_k = \sum_{i=0}^{M-1} (T_{k+i} - P_i)^2 = \sum_{i=0}^{M-1} T_{k+i}^2 - 2\sum_{i=0}^{M-1} T_{k+i}P_i + \sum_{i=0}^{M-1} P_i^2$$

则

$$X_k = 0 \iff \forall_{0 \le i < M} T_{k+i} = P_i$$

考虑通配符，把卷积改写为

$$X_k = \sum_{i=0}^{M-1} T_{k+i}^2[P_i \neq ?] - 2\sum_{i=0}^{M-1} T_{k+i}P_i + \sum_{i=0}^{M-1} [T_{k+i} \neq ?]P_i^2$$

即可。

Evaluation[8]

给出 $A_0, A_1, \cdots, A_{N-1}$, 对所有 $0 \leq k < N$，求 $f(B \cdot C^{2k} + D)$，其中

$$f(x) = \sum_{i=0}^{N-1} A_i x^i$$

$(N \leq 10^5)$

[8] HDOJ 4656

$$
\begin{aligned}
f(B \cdot C^{2k} + D) &= \sum_{i=0}^{N-1} A_i (B \cdot C^{2k} + D)^i \\
&= \sum_{i=0}^{N-1} A_i \left(\sum_{j=0}^{i} \binom{i}{j} B^j C^{2kj} D^{i-j} \right) \\
&= \sum_{i=0}^{N-1} A_i \left(\sum_{j=0}^{i} i!(j!)^{-1}[(i-j)!]^{-1} B^j C^{2kj} D^{i-j} \right) \\
&= \sum_{j=0}^{N-1} (j!)^{-1} B^j C^{2kj} D^{-j} \left(\sum_{i=j}^{N-1} A_i i! [(i-j)!]^{-1} D^i \right) \\
&= \sum_{j=0}^{N-1} (j!)^{-1} B^j C^{2kj} D^{-j} P_j
\end{aligned}
$$

计算

$$P_j = \sum_{i=j}^{N-1} A_i i! [(i-j)!]^{-1} D^i$$

构造多项式

$$U(x) = \sum_{i=0}^{N-1} A_i i! D_i x^i$$

和

$$V(x) = \sum_{i=-N-1}^{0} (-i!)^{-1} x^i$$

，则

$$(U \cdot V)(x) = \sum_{i=0}^{N-1} P_i x^i$$

$$
\begin{aligned}
&= \sum_{j=0}^{N-1} (j!)^{-1} B^j D^{-j} P_j C^{2kj} \\
&= \sum_{j=0}^{N-1} (j!)^{-1} B^j D^{-j} P_j C^{j^2+k^2-(k-j)^2} \\
&= C^{k^2} \sum_{j=0}^{N-1} [(j!)^{-1} B^j D^{-j} P_j C^{j^2}][C^{-(k-j)^2}]
\end{aligned}
$$

多项式的逆

给出多项式 $P(x)$，求 $P^{-1}(x)$ 满足

$$P(x) \cdot P^{-1}(x) \equiv 1 \pmod{x^N}$$

多项式的逆

给出多项式 $P(x)$，求 $P^{-1}(x)$ 满足

$$P(x) \cdot P^{-1}(x) \equiv 1 \pmod{x^N}$$

假设已经求出 $Q(x)$ 满足

$$P(x) \cdot Q(x) \equiv 1 \pmod{x^N}$$

要求 $P^{-1}(x)$ 满足

$$P(x) \cdot P^{-1}(x) \equiv 1 \pmod{x^{2N}}$$

$$
\begin{aligned}
& P(x)\cdot Q(x) \equiv 1 \pmod{x^N} \\
& P(x)\cdot P^{-1}(x) \equiv 1 \pmod{x^{2N}} \\
\Longrightarrow\ & P(x)\cdot (Q(x) - P^{-1}(x)) \equiv 0 \pmod{x^N} \\
\Longrightarrow\ & Q^2(x) - 2Q(x)P^{-1}(x) + P^{-2}(x) \equiv 0 \pmod{x^{2N}} \\
\Longrightarrow\ & P(x)Q^2(x) - 2Q(x) + P^{-1}(x) \equiv 0 \pmod{x^{2N}}
\end{aligned}
$$

$$
\begin{aligned}
& P(x) \cdot Q(x) \equiv 1 \pmod{x^N} \\
& P(x) \cdot P^{-1}(x) \equiv 1 \pmod{x^{2N}} \\
\Longrightarrow\ & P(x) \cdot (Q(x) - P^{-1}(x)) \equiv 0 \pmod{x^N} \\
\Longrightarrow\ & Q^2(x) - 2Q(x)P^{-1}(x) + P^{-2}(x) \equiv 0 \pmod{x^{2N}} \\
\Longrightarrow\ & P(x)Q^2(x) - 2Q(x) + P^{-1}(x) \equiv 0 \pmod{x^{2N}}
\end{aligned}
$$

时间复杂度

$$
T(N) = T(N/2) + O(N \log N) \implies T(N) = O(N \log N)
$$

城市规划[9]

求 N 个点带标号的连通无向图的数量。
$(N \le 130000)$

[9] 2013 中国国家集训队第二次作业

设 G_n 表示 n 个点带标号的无向图数量，显然

$$G_n = 2^{\binom{n}{2}}$$

设 F_n 表示 n 个点带标号的无向连通图数量，考虑点 1 所在的连通块大小，有

$$\begin{aligned}
& \sum_{k=1}^{n} \binom{n-1}{k-1} F_k G_{n-k} = G_n \\
\Longrightarrow & \sum_{k=1}^{n} \frac{F_k}{(k-1)!} \frac{G_{n-k}}{(n-k)!} = \frac{G_n}{(n-1)!} \\
\Longrightarrow & \sum_{n \geq 1} \sum_{k=1}^{n} \frac{F_k}{(k-1)!} x^k \frac{G_{n-k}}{(n-k)!} x^{n-k} = \frac{G_n}{(n-1)!} x^n \\
\Longrightarrow & \sum_{n \geq 1} \frac{F_n}{(n-1)!} x^n \equiv \frac{\sum_{n \geq 1} \frac{G_n}{(n-1)!} x^n}{\sum_{n \geq 0} \frac{G_n}{n!} x^n} \pmod{x^{N+1}}
\end{aligned}$$

多项式带余除法

多项式 $A(x), B(x)$，求 $Q(x), R(x)$ 满足

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

且 $\deg R < \deg B$。

多项式带余除法

多项式 $A(x), B(x)$，求 $Q(x), R(x)$ 满足

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

且 $\deg R < \deg B$。
设 $\deg A = n, \deg B = m$，

$$
\begin{aligned}
& A(\frac{1}{x})x^n = B(\frac{1}{x})x^m Q(\frac{1}{x})x^{n-m} + R(\frac{1}{x})x^n \\
\Longrightarrow\ & A(\frac{1}{x})x^n \equiv B(\frac{1}{x})x^m Q(\frac{1}{x})x^{n-m} \pmod{x^{n-m+1}} \\
\Longrightarrow\ & Q(\frac{1}{x})x^{n-m} \equiv A(\frac{1}{x})x^n \cdot [B(\frac{1}{x})x^m]^{-1} \pmod{x^{n-m+1}}
\end{aligned}
$$

多点求值

多项式

$$f(x) = \sum_{n=0}^{N-1} c_n x^n$$

求 $f(a_0), f(a_1), \ldots, f(a_{M-1})$。

多点求值

多项式

$$f(x) = \sum_{n=0}^{N-1} c_n x^n$$

求 $f(a_0), f(a_1), \ldots, f(a_{M-1})$。
设

$$P(x) = \prod_{i=0}^{M-1} (x - a_i)$$

$R(x) = f(x) \bmod P(x)$，则 $R(a_i) = f(a_i)$，且 $\deg R = M - 1$。

多点求值

多项式

$$f(x) = \sum_{n=0}^{N-1} c_n x^n$$

求 $f(a_0), f(a_1), \ldots, f(a_{M-1})$。
设

$$P(x) = \prod_{i=0}^{M-1} (x - a_i)$$

$R(x) = f(x) \bmod P(x)$，则 $R(a_i) = f(a_i)$，且 $\deg R = M - 1$。
分治 $O(N \log^2 N)$

常系数线性递推关系

当 $n \geq K$ 时，有

$$a_n = \sum_{i=1}^{K} c_i a_{n-i}$$

给出 $a_0, a_1, \ldots, a_{K-1}$，求 a_N。

常系数线性递推关系

当 $n \geq K$ 时，有

$$a_n = \sum_{i=1}^{K} c_i a_{n-i}$$

给出 $a_0, a_1, \ldots, a_{K-1}$，求 a_N。
令

$$\mathbf{x}_n = \begin{bmatrix} a_n \\ a_{n+1} \\ \cdots \\ a_{n+K-1} \end{bmatrix}, \mathrm{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ c_K & c_{K-1} & c_{K-2} \cdots & c_1 & \end{bmatrix}$$

则 $\mathbf{x}_N = \mathbf{A}^N \mathbf{x}_0$。

对于任意 $n \geq 0$，$\mathbf{A}^n\mathbf{x}_0$ 可以表示为 $\mathbf{A}^0\mathbf{x}_0, \mathbf{A}^1\mathbf{x}_0, \ldots, \mathbf{A}^{K-1}\mathbf{x}_0$ 的线性组合。

对于任意 $n \geq 0$，$\mathbf{A}^n\mathbf{x}_0$ 可以表示为 $\mathbf{A}^0\mathbf{x}_0, \mathbf{A}^1\mathbf{x}_0, \ldots, \mathbf{A}^{K-1}\mathbf{x}_0$ 的线性组合。
设

$$\mathbf{A}^n\mathbf{x}_0 = \sum_{i=0}^{K-1} a_i \mathbf{A}^i \mathbf{x}_0$$

$$\mathbf{A}^m\mathbf{x}_0 = \sum_{i=0}^{K-1} b_i \mathbf{A}^i \mathbf{x}_0$$

则

$$\begin{aligned}
\mathbf{A}^{n+m}\mathbf{x}_0 &= \mathbf{A}^n(\sum_{j=0}^{K-1} b_j \mathbf{A}^j \mathbf{x}_0) \\
&= \sum_{j=0}^{K-1} b_j \mathbf{A}^j (\sum_{i=0}^{K-1} a_i \mathbf{A}^i \mathbf{x}_0) \\
&= \sum_{k=0}^{2K-2} (\sum_{i=0}^{K-1} a_i b_{k-i}) \mathbf{A}^k \mathbf{x}_0)
\end{aligned}$$

$K-1$ 次多项式的乘积是 $2K-2$ 次多项式，利用

$$\mathbf{A}^K\mathbf{x}_0 = \sum_{i=1}^{K} c_i \mathbf{A}^{K-i}\mathbf{x}_0$$

于是只需计算

$$(\sum_{i=0}^{K-1} a_i z^i)(\sum_{j=0}^{K-1} b_j z^j) \bmod (z^K - \sum_{i=0}^{K-1} c_{K-i} z^i)$$

$K-1$ 次多项式的乘积是 $2K-2$ 次多项式，利用

$$\mathbf{A}^K\mathbf{x}_0=\sum_{i=1}^{K}c_i\mathbf{A}^{K-i}\mathbf{x}_0$$

于是只需计算

$$(\sum_{i=0}^{K-1}a_iz^i)(\sum_{j=0}^{K-1}b_jz^j) \bmod (z^K-\sum_{i=0}^{K-1}c_{K-i}z^i)$$

倍增求 $\mathbf{A}^N\mathbf{x}_0$ 的表示 $O(K\log K\log N)$
设

$$\mathbf{A}^N\mathbf{x}_0=\sum_{i=0}^{K-1}k_i\mathbf{A}^i\mathbf{x}_0$$

则

$$a_n=\sum_{i=0}^{K-1}k_ia_i$$

多项式平方根

多项式 $P(x)$，求 $Q(x)$ 满足

$$Q^2(x) \equiv P(x) \pmod{x^N}$$

多项式平方根

多项式 $P(x)$，求 $Q(x)$ 满足

$$Q^2(x) \equiv P(x) \pmod{x^N}$$

假设

$$Q^2(x) \equiv P(x) \pmod{x^N}$$

则

$$\begin{aligned}
& (Q^2(x) - P(x))^2 \equiv 0 \pmod{x^{2N}} \\
\Longrightarrow\ & (Q^2(x) + P(x))^2 \equiv 4Q^2(x)P(x) \pmod{x^{2N}} \\
\Longrightarrow\ & \left(\frac{Q^2(x) + P(x)}{2Q(x)}\right)^2 \equiv P(x) \pmod{x^{2N}}
\end{aligned}$$

$$\left(\frac{Q^2(x)+P(x)}{2Q(x)}\right)^2 \equiv P(x) \pmod{x^{2N}}$$

注意到

$$\left(\frac{Q^2(x)+P(x)}{2Q(x)}\right)^{-1} \equiv Q(x)^{-1} \pmod{x^{N}}$$

逆元可以快速维护，时间复杂度 $O(N\log N)$

The Child and Binary Tree[10]

给出 $c_1, c_2, \ldots, c_N$，令 f_n 满足

$$f_n = \begin{cases} \sum_{i=1}^{N} \sum_{k=0}^{n-c_i} f_k \cdot f_{n-k} & n > 0 \\ 1 & n = 0 \end{cases}$$

求 $f_1, f_2, \ldots, f_M$ 的值。

[10] Codeforces Round #250

The Child and Binary Tree[10]

给出 $c_1, c_2, \ldots, c_N$，令 f_n 满足

$$f_n = \begin{cases} \sum_{i=1}^{N} \sum_{k=0}^{n-c_i} f_k \cdot f_{n-k} & n > 0 \\ 1 & n = 0 \end{cases}$$

求 $f_1, f_2, \ldots, f_M$ 的值。
考虑生成函数 $C(x) = \sum_{i=1}^{N} x^{c_i}$，$F(x) = \sum_{n \geq 0} f_n x^n$ 则

$$F(x) = 1 + F^2(x) \cdot C(x)$$

解得

$$F(x) = \frac{1 - \sqrt{1 - 4C(x)}}{2C(x)}$$

[10] Codeforces Round #250

$$f_n = \sum_{i=1}^{N} \sum_{k=0}^{n-c_i} f_k \cdot f_{n-k}$$

分治，考虑 $f_l, f_{l+1}, \ldots, f_{m-1}$ 对 $f_m, f_{m+1}, \ldots, f_{r-1}$ 的贡献。
只需考虑 $F(x)$ 和 $C(x)$ 的前 $r-l$ 项，时间复杂度

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

Nim[11]

统计满足下列条件的 $(x_1, x_2, \ldots, x_N)$ 数量：

- $1 \le x_i \le K$
- x_i 是质数
- $x_1 \oplus x_2 \oplus \cdots x_N = 0$

$(N \le 10^9, K \le 50000)$

[11] Single Round Match 518, Level 3

Fast Walsh transform

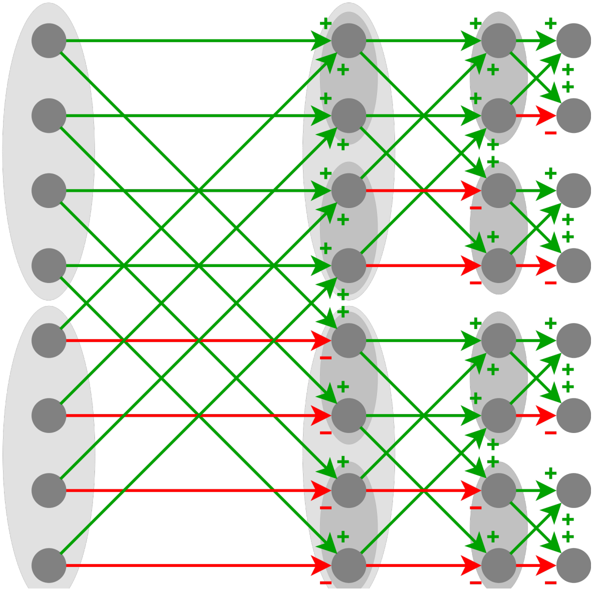
对于给出的 $a_0, a_1, \ldots, a_{2^n-1}$，考虑变换

$$x_k = \sum_{i=0}^{2^n-1} a_i(-1)^{|i \wedge k|}$$

考虑 $b_0, b_1, \ldots, b_{2^n-1}$

$$y_k = \sum_{j=0}^{2^n-1} b_j(-1)^{|j \wedge k|}$$

$$\begin{aligned} x_k y_k &= (\sum_{i=0}^{2^n-1} a_i(-1)^{|i \wedge k|})(\sum_{j=0}^{2^n-1} b_j(-1)^{|j \wedge k|}) \\ &= \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} a_i b_j (-1)^{|(i \oplus j) \wedge k|} \\ &= \sum_{i=0}^{2^n-1} (\sum_{j=0}^{2^n-1} a_i b_{i \oplus j})(-1)^{|i \wedge k|} \end{aligned}$$



计算向量 $\mathbf{w}_n = (w_0, w_1, \ldots, w_{2^k-1})$ 表示 n 个变量，有 w_i 种方法异或和是 i。
倍增地计算向量 $\mathbf{w}_n$，时间复杂度 $O(K \log K \log N)$

祝大家省选顺利！