快速傅里叶变换

孙科

北京师范大学

主要内容

Outline

1. 快速傅里叶变换
   - 准备工作
   - 推导过程
2. 快速数论变换
3. 练习题目
   - 卷积应用
   - 高维 FFT
   - 分治 FFT
   - 对时间分治的 FFT
   - 倍增 FFT

问题的提出

- 普通的高精度乘法时间复杂度？

问题的提出

- 普通的高精度乘法时间复杂度？
- $O((\frac{N}{C})^2)$
- 如何优化？

多项式

一个以 x 为变量的多项式定义在一个代数域 F 上，将函数 $A(x)$ 表示为形式和:

$$A(x) = \sum_{j=0}^{n-1} a_j x^j$$

多项式

一个以 x 为变量的多项式定义在一个代数域 F 上，将函数 $A(x)$ 表示为形式和:

$$A(x) = \sum_{j=0}^{n-1} a_j x^j$$

我们称 $a_0, a_1, \ldots, a_{n-1}$ 为如上多项式的系数，所有系数都属于域 F，典型的情形是复数集合 C。

多项式

一个以 x 为变量的多项式定义在一个代数域 F 上，将函数 $A(x)$ 表示为形式和:

$$A(x) = \sum_{j=0}^{n-1} a_j x^j$$

我们称 $a_0, a_1, \ldots, a_{n-1}$ 为如上多项式的系数，所有系数都属于域 F，典型的情形是复数集合 C。

如果一个多项式 $A(x)$ 的最高次的非零系数是 a_k，则称 $A(x)$ 的次数是 k，记 $degree(A) = k$。任何严格大于一个多项式次数的整数都是该多项式的**次数届**，因此，对于次数届为 n 的多项式，其次数可以是 $0 \sim n-1$ 之间的任何整数。

多项式加法

如果 $A(x)$ 和 $B(x)$ 是次数届为 n 的多项式，那么它们的和也是一个次数届为 n 的多项式 $C(x)$，对所有属于定义域的 x，都有 $C(x) = A(x) + B(x)$。也就是说，
若

$$A(x) = \sum_{j=0}^{n-1} a_j x^j \qquad B(x) = \sum_{j=0}^{n-1} b_j x^j$$

则

$$C(x) = \sum_{j=0}^{n-1} c_j x^j (c_j = a_j + b_j)$$

多项式加法

如果 $A(x)$ 和 $B(x)$ 是次数届为 n 的多项式，那么它们的和也是一个次数届为 n 的多项式 $C(x)$，对所有属于定义域的 x，都有 $C(x)=A(x)+B(x)$。也就是说，
若

$$A(x)=\sum_{j=0}^{n-1}a_jx^j \qquad B(x)=\sum_{j=0}^{n-1}b_jx^j$$

则

$$C(x)=\sum_{j=0}^{n-1}c_jx^j(c_j=a_j+b_j)$$

例如，如果有多项式 $A(x)=6x^3+7x^2-10x+9$ 和 $B(x)=-2x^3+4x-5$，那么 $C(x)=4x^3+7x^2-6x+4$。

多项式乘法

多项式乘法如果 $A(x)$ 和 $B(x)$ 是次数届为 n 的多项式，那么它们的乘积 $C(x)$ 是一个次数届为 $2n-1$ 的多项式 $C(x)$，对所有属于定义域的 x，都有 $C(x) = A(x)B(x)$。
形式化的式子有

$$C(x) = \sum_{j=0}^{2n-2} c_j x^j$$

其中

$$c_j = \sum_{k=0}^{j} a_k b_{j-k}$$

多项式乘法

此时

$$degree(C) = degree(A) + degree(B)$$

用之前的例子，将得到

$$C(x) = -12x^6 - 14x^5 + 44x^4 - 20x^3 - 75x^2 + 86x - 45$$

多项式表示

从某种意义上，多项式的系数表达与点值表达式等价的。

系数表达

对一个次数届为 n 的多项式 $A(x)=\sum_{j=0}^{n-1} a_j x^j$ 而言，其系数表达是一个由系数组成的（列）向量 $a=(a_0,a_1,\ldots,a_{n-1})$。

系数表达

对一个次数届为 n 的多项式 $A(x)=\sum_{j=0}^{n-1}a_jx^j$ 而言，其系数表达是一个由系数组成的（列）向量 $a=(a_0,a_1,\ldots,a_{n-1})$。

对于多项式乘法，系数向量 c 成为输入向量 a 和 b 的卷积，表示成 $c=a\otimes b$。

卷积

卷积形式：

$$c = a \otimes b$$

$$c_j = \sum_{k=0}^{j} a_k b_{j-k}$$

卷积形式：

$$c = a \otimes b$$

$$c_j = \sum_{k=0}^{j} a_k b_{j-k}$$

卷积运算性质：

卷积

卷积形式：

$$c = a \otimes b$$

$$c_j = \sum_{k=0}^{j} a_k b_{j-k}$$

卷积运算性质：

- 交换律：$a \otimes b = b \otimes a$

卷积

卷积形式：

$$c = a \otimes b$$

$$c_j = \sum_{k=0}^{j} a_k b_{j-k}$$

卷积运算性质：

- 交换律：$a \otimes b = b \otimes a$
- 结合律：$(a \otimes b) \otimes c = a \otimes (b \otimes c)$

卷积

卷积形式：

$$c = a \otimes b$$

$$c_j = \sum_{k=0}^{j} a_k b_{j-k}$$

卷积运算性质：

- 交换律：$a \otimes b = b \otimes a$
- 结合律：$(a \otimes b) \otimes c = a \otimes (b \otimes c)$
- 分配律：$(a + b) \otimes c = a \otimes c + b \otimes c$

点值表达

一个次数届为 n 的多项式 $A(x)$ 的点值表达就是一个由 n 个点值对组成的集合 $\{(x_0, y_0), (x_1, y_1) \ldots (x_{n-1}, y_{n-1})\}$ 使得对 $k = 0, 1, \ldots, n-1$，所有 x_k 各不相同，且 $y_k = A(x_k)$。

点值表达

一个次数届为 n 的多项式 $A(x)$ 的点值表达就是一个由 n 个点值对组成的集合 $\{(x_0, y_0), (x_1, y_1) \ldots (x_{n-1}, y_{n-1})\}$ 使得对 $k = 0, 1, \ldots, n-1$，所有 x_k 各不相同，且 $y_k = A(x_k)$。

一个多项式可以有很多不同的点值表达。如果两个多项式采用的点相同的话，用点值表达多项式做乘法只需 $O(n)$ 的时间。

求值与插值

从一个多项式的系数表达转化为点值表达的过程是求值，其逆运算称为插值。

从一个多项式的系数表达转化为点值表达的过程是求值，其逆运算称为插值。
拉格朗日插值：

$$A(x) = \sum_{k=0}^{n-1} y_k \frac{\prod_{j\neq k}(x - x_j)}{\prod_{j\neq k}(x_k - x_j)}$$

定理

插值多项式的唯一性定理：对于任意 n 个点值对组成的集合 $\{(x_0, y_0), (x_1, y_1) \ldots (x_{n-1}, y_{n-1})\}$，其中所有的 x_k 都不同，那么存在唯一的次数届为 n 的多项式 $A(x)$，满足 $y_k = A(x_k)$。

求值与插值

简单的求值和插值的时间复杂度都是 $O(n^2)$ 的。我们之后就要通过巧妙选取点来加速这两个过程，使运行时间变为 $O(nlogn)$。

什么是复数

定义

把形如 $z = a + bi$(a，b 均为实数) 的数称为复数，其中 a 称为实部，b 称为虚部，i 称为虚数单位。

什么是复数

定义

把形如 $z = a + bi$(a，b 均为实数) 的数称为复数，其中 a 称为实部，b 称为虚部，i 称为虚数单位。

关于虚数单位 i：$i^2 = -1$

什么是复数

定义

把形如 $z = a + bi$(a，b 均为实数) 的数称为复数，其中 a 称为实部，b 称为虚部，i 称为虚数单位。

关于虚数单位 i：$i^2 = -1$
复数域是对实数域的扩展，从一维到二维

复数运算

- 考虑两个复数 $z_1 = a + bi, z_2 = c + di$:
  - $z_1 + z_2 = (a + c) + (b + d)i$
  - $z_1 \times z_2 = (ac - bd) + (ad + bc)i$

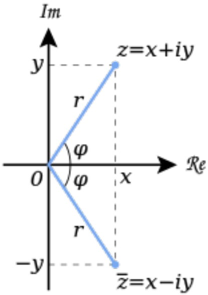
复数运算

- 考虑两个复数 $z_1 = a + bi, z_2 = c + di$:
  - $z_1 + z_2 = (a + c) + (b + d)i$
  - $z_1 \times z_2 = (ac - bd) + (ad + bc)i$
- 复数的模：$|z_1| = \sqrt{a^2 + b^2}$

复数运算

- 考虑两个复数 $z_1 = a + bi, z_2 = c + di$:
  - $z_1 + z_2 = (a + c) + (b + d)i$
  - $z_1 \times z_2 = (ac - bd) + (ad + bc)i$
- 复数的模：$|z_1| = \sqrt{a^2 + b^2}$
- 共轭复数：两个复数实部相同，虚部互为相反数

复数的几何意义



$$z = r(\cos\phi + i\sin\phi)$$

重要公式

前方高能!!!

重要公式

前方高能！！！

棣莫弗定理

$$\big(\cos(x) + i\sin(x)\big)^n = \cos(nx) + i\sin(nx)$$

震惊，复数乘法的本质竟然是旋转...?!

前方高能!!!

棣莫弗定理

$$\big(\cos(x) + i\sin(x)\big)^n = \cos(nx) + i\sin(nx)$$

震惊，复数乘法的本质竟然是旋转...?!

欧拉公式

$$e^{ix} = \cos x + i\sin x$$
$$e^{i\pi} + 1 = 0$$

最基本的 5 个数学常数之间的联系如此简洁!

单位复数根

n 次单位复数根是满足 $\omega^n = 1$ 的复数 ω。
n 次单位复数根恰好有 n 个：

$$\omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1}$$

单位复数根

n 次单位复数根是满足 $\omega^n = 1$ 的复数 ω。
n 次单位复数根恰好有 n 个：

$$\omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1}$$

其中**主 n 次单位复数根**为

$$\omega_n = e^{2\pi i/n} = \cos(2\pi/n) + i\sin(2\pi/n)$$

其他 n 次单位复数根都是 ω_n 的幂次。

准备工作

消去引理

对于任何整数 $n \geq 0, k \geq 0, d > 0$, 有 $\omega_{dn}^{dk} = \omega_n^k$

消去引理推论

对于任意偶数 $n > 0$, 有 $\omega_n^{n/2} = \omega_2 = -1$

准备工作

折半引理

如果 $n > 0$ 为偶数，那么 n 个 n 次单位复数根的平方的集合就是 $n/2$ 个 $n/2$ 次单位复数根的集合。

求和引理

对任意整数 $n \geq 1$ 和不能被 n 整除的非负整数 k，有 $\sum_{j=0}^{n-1} (\omega_n^k)^j = 0$。

Outline

DFT

现在我们希望计算次数届 n 的多项式

$$A(x)=\sum_{j=0}^{n-1} a_j x^j$$

在 ω_n^k 处的值，记为 y_k

$$y_k = A(\omega_n^k) = \sum_{j=0}^{n-1} a_j \omega_n^{kj}$$

DFT

现在我们希望计算次数届 n 的多项式

$$A(x) = \sum_{j=0}^{n-1} a_j x^j$$

在 ω_n^k 处的值，记为 y_k

$$y_k = A(\omega_n^k) = \sum_{j=0}^{n-1} a_j \omega_n^{kj}$$

向量 $y = (y_0, y_1, \ldots, y_{n-1})$ 就是系数向量 $a = (a_0, a_1, \ldots, a_{n-1})$ 的离散傅里叶变换（DFT），记为 $y = DFT_n(a)$。

FFT

快速傅里叶变换（FFT）利用复数单位根的特殊性质，可以在 $O(n\log n)$ 时间内计算出 $DFT_n(a)$。

快速傅里叶变换（FFT）利用复数单位根的特殊性质，可以在 $O(n \log n)$ 时间内计算出 $DFT_n(a)$。

为方便分治计算，这里 n 必须恰好是 2 的整数幂。

FFT

采用 $A(x)$ 中偶数下标的系数与奇数下标的系数，分别定义两个新的次数届为 $n/2$ 的多项式 $A^{[0]}(x)$ 和 $A^{[1]}(x)$:

$$A^{[0]}(x) = a_0 + a_2 x + a_4 x^2 + \cdots + a_{n-2} x^{n/2-1}$$

$$A^{[1]}(x) = a_1 + a_3 x + a_5 x^2 + \cdots + a_{n-1} x^{n/2-1}$$

FFT

采用 $A(x)$ 中偶数下标的系数与奇数下标的系数，分别定义两个新的次数届为 $n/2$ 的多项式 $A^{[0]}(x)$ 和 $A^{[1]}(x)$:

$$A^{[0]}(x) = a_0 + a_2 x + a_4 x^2 + \cdots + a_{n-2} x^{n/2-1}$$

$$A^{[1]}(x) = a_1 + a_3 x + a_5 x^2 + \cdots + a_{n-1} x^{n/2-1}$$

于是有

$$A(x) = A^{[0]}(x^2) + xA^{[1]}(x^2)$$

所以，求 $A(x)$ 在 $\omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1}$ 处的值转换为求次数届为 $n/2$ 的多项式 $A^{[0]}(x)$ 和 $A^{[1]}(x)$ 在点 $(\omega_n^0)^2, (\omega_n^1)^2, \ldots, (\omega_n^{n-1})^2$ 的值。可以发现其实是 $n/2$ 个 $n/2$ 次单位复数根，且每个根恰好出现两次。

将点值表达的多项式转换回系数表达，是相似的过程。

下面是简要推导：
我们把 DFT 写成矩阵乘积 $y = V_n a$。其中 V_n 是一个范德蒙德矩阵，在 (k, j) 处的元素为 ω_n^{kj}。对于逆运算 $a = DFT_n^{-1}(y)$，我们把 y 左边乘以 V_n 的逆矩阵来处理。

定理

对 $j, k = 0, 1, \ldots, n-1$，V_n^{-1} 在 (j, k) 元素为 ω_n^{-kj}/n。

证明 $V_n^{-1} V_n = I_n$ 时用求和引理即可，注意使用条件。

IDFT

可以推导出 $DFT_n^{-1}(y)$：

$$a_j = \frac{1}{n}\sum_{k=0}^{n-1} y_k \omega_n^{-kj}$$

可以看出只需将单位根取倒数，做一次 FFT，最后将结果都除以 n，就做完逆变换了。

代码实现

试着写写代码（递归/非递归）

- 复数类
- 二进制反转
- 蝴蝶操作

NTT

可以看到 FFT 的缺陷在于用到了浮点数进行运算，考虑两个次数为 N，系数不超过 P 的多项式相乘，那么结果最大会到 NP^2 的级别，如 $N = 10^5, P = 10^9$，此时 double 的有效位数已经不够了，会造成误差

于是可以用快速数论变换 (NTT)，NTT 同样是用来加速多项式乘法的，优势就在于是在模意义下计算，可以对某些质数取模

原根

若 $m > 1, (a, m) = 1$，则使得同余式

$$a^d \equiv 1 \pmod{m}$$

成立的最小正整数 d 叫做 a 对模 m 的**指数**

若 a 对模 m 的指数是 $\varphi(m)$，则 a 叫做模 m 的一个**原根**

一些性质及定理

- 若 a 对模 m 的指数是 d，则 $a^0, a^1, \ldots, a^{d-1}$ 对模 m 两两不同余
- 若 a 对模 m 的指数是 d，则 $d \mid \varphi(m)$
- 模 m 有原根的充要条件:$m = 2, 4, p^e, 2 \times p^e (p$ 是奇质数$)$

NTT

设 g 是模数 P 的原根，记 g_n 为 $g^{\frac{P-1}{n}} \pmod P$，会发现与 FFT 中的 $\omega_n = e^{2\pi i/n}$ 有相似的性质。

消去引理

$$g_{dn}^{dk} \equiv g_n^k \pmod P$$

$$g_n^{n/2} \equiv -1 \pmod P$$

折半引理

$$(g_n^{k+n/2})^2 \equiv g_n^{2k+n} \equiv (g_n^k)^2 \pmod P$$

求和引理

$$\sum_{j=0}^{n-1} (g_n^k)^j \equiv 0 \pmod P$$

NTT 模数

使用条件：模数 P 为质数且 n 必须是 $P-1$ 的因数，因为 n 是 2 的幂，所以 P 形如 $c \cdot 2^k + 1$

一些 NTT 模数：

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

任意模数 NTT

模数 P 如果不满足上述条件怎么办?

可以取几个满足条件的 NTT 模数比如 $p_1, p_2, p_3, \ldots$，使得乘积大于 nP^2，分别做 NTT，之后用中国剩余定理合并。

Outline

多项式乘法

- 先赠送两个模板题，请随意 AC
- loj 108 多项式乘法
- bzoj 2179 FFT 快速傅立叶

快速傅立叶之二[1]

给定两个长度为 n 的序列 a 和 b，计算
$C_k = \sum_{i=k}^{n-1} a_i b_{i-k} (0 \leq k < n)$。$n \leq 100000, a_i \leq 100, b_i \leq 100$

[1] bzoj 2194

快速傅立叶之二[1]

给定两个长度为 n 的序列 a 和 b，计算
$C_k = \sum_{i=k}^{n-1} a_i b_{i-k} (0 \le k < n)$。$n \le 100000, a_i \le 100, b_i \le 100$

发现将 a 数组反转后，原式就变成了卷积形式，
$C_k = \sum_{i=k}^{n-1} a_{n-i-1} b_{i-k} (0 \le k < n)$，最后再倒序输出结果即可。

[1] bzoj 2194

[Zjoi2014] 力[2]

给出 n 个数 q_i，给出 F_j 的定义如下：

$$F_j = \sum_{i<j} \frac{q_i q_j}{(i-j)^2} - \sum_{i>j} \frac{q_i q_j}{(i-j)^2}$$

令 $E_i = \frac{F_i}{q_i}$，试求 E_i。

[2] bzoj 3527

减号左右两边分别求解，左边：

$$a_i = q_i, b_i = \frac{1}{i^2}, c_i = \sum_{j<i} \frac{q_j}{(i-j)^2}$$

则 $c = a \otimes b$

右边仿照上一题，将 a 翻转之后化成卷积形式

idiots[3]

给定 n 个长度分别为 a_i 的木棒，问随机选择 3 个木棒能够拼成三角形的概率。$N \leq 10^5, a_i \leq 10^5$。

[3] bzoj 3513,hdu 4609

- 构成三角形条件：任意两边之和大于第三边
- 统计出长度为 i 的木棒个数 num_i，作为多项式系数
- 将该多项式自己与自己卷积得到选两个木棒的方案数，然后扫一遍统计答案，需要去掉重复

Outline

二维 FFT

二维卷积：

$$C_{x,y} = \sum_{0 \le i < x, 0 \le j < y} A_{i,j} B_{x-i,y-j}$$

二维离散傅里叶变换：

$$y_{k_1,k_2} = \sum_{j_1=0}^{n_1-1} \sum_{j_2=0}^{n_2-1} a_{j_1,j_2} \omega_{n_1}^{j_1 k_1} \omega_{n_2}^{j_2 k_2}$$

求法：沿着第一维计算 n/n_1 个独立的一维 DFT，然后把 DFT 的结果作为输入，再计算沿着第二维的算 n/n_2 个独立的一维 DFT，即得到二维 DFT 的结果，更高维度同理。

- 计算维度的次序不影响结果
- 如果用 FFT 计算每一维 DFT，那么总时间复杂度为 $O(n \log n)$，其中 n 为各个维度大小的乘积，与维度数无关。

Five Dimensional DFT[4]

给一个五维、各个维度大小不超过 10 的复数数组 a，求五维 DFT：

$$A[j_1][j_2][j_3][j_4][j_5] = \sum_{i_1=0}^{n_1-1} \cdots \sum_{i_5=0}^{n_5-1} a[j_1][j_2][j_3][j_4][j_5]\, e^{2\pi i(i_1 j_1/n_1+\cdots+i_5 j_5/n_5)}$$

[4]2017 ICPC 南宁赛区 G 题

Five Dimensional DFT[4]

给一个五维、各个维度大小不超过 10 的复数数组 a，求五维 DFT：

$$A[j_1][j_2][j_3][j_4][j_5] = \sum_{i_1=0}^{n_1-1} \cdots \sum_{i_5=0}^{n_5-1} a[j_1][j_2][j_3][j_4][j_5]\, e^{2\pi i(i_1 j_1/n_1+\cdots+i_5 j_5/n_5)}$$

做法：这里各维度大小比较小，做 DFT 时没必要用 FFT，暴力做即可，时间复杂度 $O(n^6)$。

[4] 2017 ICPC 南宁赛区 G 题

Outline

第一类斯特林数

把 N 个人安排到 K 张圆桌上，求有几种方案。圆桌没有顺序。一张圆桌上的人有按逆时针排列的顺序。

实际上就是第一类斯特林数，记作 $s(n,k)$

- $s(n,0)=0, s(1,1)=1$
- $s(n,k)=s(n-1,k-1)+(n-1)s(n-1,k)$
- $\sum_{k=0}^{n} s(n,k)=n!$

分治 FFT

- 记 x 的 n 次上升幂为 $x^{\overline{n}}$
- $x^{\overline{n}} = \prod_{i=0}^{n-1}(x+i) = \sum_{i=0}^{n-1} s(n,i)x^i$
- 考虑分治 FFT，$work(l,r) = work(l,mid) \times work(mid,r)$
- 也可以理解为对这 n 个多项式用 FFT 做启发式合并
- 时间复杂度 $O(n\log^2 n)$

Outline

给出长度分别为 1 到 n 的珠子，长度为 i 的珠子有 $g[i]$ 种，每种珠子有无限个，问用这些珠子串成长度为 n 的链有多少种方案。
$n \leq 100000$, 答案模 313。

[5] hdu 5730

Shell Necklace

dp 式子非常简单，记 f[n] 表示串成长度为 n 的方案数，转移为:

$$f_n = \sum_{i=1}^{n} g_i f_{n-i}$$

发现是个卷积形式，但与正常的卷积不同！计算第 n 项时需要先求出前面所有项。

dp 式子非常简单，记 f[n] 表示串成长度为 n 的方案数，转移为:

$$f_n = \sum_{i=1}^{n} g_i f_{n-i}$$

发现是个卷积形式，但与正常的卷积不同！计算第 n 项时需要先求出前面所有项。

采用 cdq 分治的思想，对于 $[l, r]$ 范围的 dp 值，我们先计算出 $[l, mid]$ 的值，然后把这一部分的贡献加到 $[mid + 1, r]$，最后递归计算 $[mid + 1, r]$ 的 dp 值。时间复杂度 $O(n \log^2 n)$。

对时间分治

设 $mid+1 \le x \le r$, 贡献

$$d[x] = \sum_{i=l}^{mid} f[i]g[x-i]$$

为了方便，范围扩大到 $[l, x-1]$，此时 $[mid+1, r]$ 这段 f 值都为 0：

$$d[x] = \sum_{i=l}^{x-1} f[i]g[x-i]$$

平移一下，设 $a[i] = f[i+l], b[i] = g[i+1], x' = x-l$

$$d[x] = \sum_{i=0}^{x-l-1} a[i]b[x-l-1-i]$$

$$d[x'] = \sum_{i=0}^{x'-1} a[i]b[x'-1-i]$$

Outline

[JSOI2012] 分零食[6]

有 N 个小朋友排成一列，要分 M 个糖果，要求没有分到糖果的必须是最后的连续若干位。如果一个小朋友没有分到糖果，那么欢乐程度是 1。如果一个小朋友分到了 x 个糖果，那么欢乐程度是 $f(x) = Ox^2 + Sx + U$。求所有方案下欢乐程度乘积的总和，答案模 P。$M \leq 10^4, N \leq 10^8, P \leq 255$。

[6] luogu P5075

[JSOI2012] 分零食

构造多项式

$$F(x) = f(1)x + f(2)x^2 + \cdots + f(m)x^m = \sum_{i=1}^{m} f(i)x^i$$

不难发现 $F^k(x)$ 的 x^m 项系数就是所有方案中发给前 k 个小朋友糖果的乘积之和所以答案为

$$G(x) = F(x) + F^2(x) + \cdots + F^n(x)$$

的 x^m 项系数

之后倍增即可，类似poj 3233 Matrix Power Series

[JSOI2012] 分零食

记 $G_n(x) = \sum_{i=1}^{n} F^i(x)$

当 n 是偶数时

$$G_n(x) = G_{n/2}(x) + F^{n/2}(x)\,G_{n/2}(x) = (1 + F^{n/2}(x))\,G_{n/2}(x)$$

$$F^n(x) = (F^{n/2}(x))^2$$

当 n 是奇数时 $G_n(x)$ 再加上个 $F^n(x)$,$F^n(x)$ 再多乘一次 $F(x)$ 即可

Homework

- loj 108 多项式乘法
- bzoj 3527 [Zjoi2014] 力
- spoj Triple Sums
- Hong Kong Regional Online Preliminary 2016 A+B Problem
- luogu P4173 残缺的字符串
- luogu P4721 【模板】分治 FFT