Trivial Polynomial

___debug

2017 年 1 月 5 日

回顾

FFT

- 设 $F(x)=F_0(x^2)+xF_1(x^2)$
- $F(\omega_n^i)=F_0(\omega_n^{2i})+\omega_n^iF_1(\omega_n^{2i})=F_0(\omega_{n/2}^i)+\omega_n^iF_1(\omega_{n/2}^i)$
- $F(\omega_n^{i+n/2})=F(-\omega_n^i)=F_0(\omega_{n/2}^i)-\omega_n^iF_1(\omega_{n/2}^i)$
- (IDFT) $[x^i]F=\frac{1}{n}\sum_{j=0}^{n-1}F(\omega_n^j)\omega_n^{-ij}$

NTT

令 P 的原根为 g, 用 $g^{\frac{P-1}{n}}$ 代替 ω_n 即可.
如果模数不是 998244353, 怎么求原根?
其实暴力验证也是可以的:)

之后即使实际上要用 NTT, 一般也会写成 FFT.

生成函数

- 一般生成函数 (OGF):

$$f(x)=\sum_{k=0}^{n} a_k x^k$$

- 指数生成函数 (EGF):

$$f(x)=\sum_{k=0}^{n} \frac{a_k}{k!} x^k$$

前者常用于解决组合问题，后者常要用于解决排列问题.
原因？
把两个 EGF 乘起来，发现正好凑出了一个排列数的形式.
先做几道题找找感觉.

Codeforces 286E, Ladies' Shop

给你一个长度为 N 的序列 $\{a_i\}$ 和一个正整数 M，求一个长度最小的序列 $\{p_i\}$，满足对其做完全背包，能凑出的大小 $\leq M$ 的物品组成的序列恰好为 $\{a_i\}$.

要求输出方案，如果无解输出 NO.

$N, M \leq 10^6$

首先如果有解, 对于每个 a_i 一定只有以下两种情况:

- 不能被 $< a_i$ 的任意一些 a_j 表示
- 能被 $< a_i$ 的**至少两个** a_j 表示 (也就是说虽然可能被多个 a_j 表示, 但是一定存在 $a_j + a_k = a_i$)

上面的第二个条件的原因是, 如果能被 $< a_i$ 的表示但至少需要大于两个 a_j, 那么就无解了.

然后我们发现已经将 "很多个数加起来" 转化为了 "两个数加起来", 于是设 $\{a_i\}$ 的生成函数为 A, 那么只要求出 A^2, 我们有:

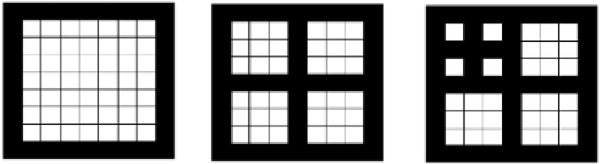
- 若 $[x^i]A = 1$ 且 $[x^i]A^2 = 0$, 则 i 一定在答案中
- 若 $[x^i]A = 0$ 且 $[x^i]A^2 = 1$, 无解

FFT 即可.

时间复杂度 $O(N + M \log M)$.

Codeforces 300D, Painting Square

给出一个 $N \times N$ 的矩形, 每次可以这么等分:



[1]

(显然 N 必须为奇数才可以继续分下去)

Q 次询问, 每次给定 N, K, 问将 $N \times N$ 做 K 次操作最终有多少种情况. (不考虑操作顺序不同)

对 7340033 取模.

$N \leq 10^9, K \leq 1000, Q \leq 10^5$

[1] An example of correct painting at n = 7 & k = 2.

首先考虑暴力 DP.
当 N 为奇数并且 > 1 时，显然有

$$dp(n,k) = \sum_{a+b+c+d=k-1} dp(\frac{n-1}{2},a)dp(\frac{n-1}{2},b)dp(\frac{n-1}{2},c)dp(\frac{n-1}{2},d)$$

发现 $a+b+c+d=k-1$ 这个部分很像卷积的形式，生成函数?
设 $F(n)$ 表示 $dp(n,0...k)$ 的生成函数，显然有

$$F(n) = F(\frac{n-1}{2})^4 x + 1$$

最后乘上一个 x 是把多项式整体右移一位，然后 $k=0$ 时答案为 1 还得加上去.
但是 Q,N 这么大，不能直接做啊?

仔细观察发现，N 是一直除 2 的，如果对于两个 n_1,n_2，将它们一直除 2 直到 $n \bmod 2 = 0$ 或是 $n \le 1$ 所需要的除 2 次数相同，那么 n_1,n_2 其实是完全一样的.
所以我们只需处理 $\log N$ 个多项式了.
时间复杂度 $O(K\log K\log N + Q\log N)$.

接下来介绍一些多项式的基本运算.

多项式求导 & 积分

Interesting.

多项式求逆

对于给定的 $A(x)$，求出 $B(x)$，满足

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

经过后面的推导可以看出，只要 $[x^0]A$ 可逆，那么 $A(x)$ 也一定可逆.

考虑分治. 对于这种多项式的分治, 有一个很重要的性质可以利用:

$$\begin{aligned} F(x) &\equiv 0 \pmod{x^n} \\ \Rightarrow F(x)^2 &\equiv 0 \pmod{x^{2n}} \end{aligned}$$

注意如果右边不是 0 了, 那么这个性质就不成立了.

现在假设我们已经求出了 $A(x)B'(x) \equiv 1 \pmod{x^{\lceil n/2 \rceil}}$.
推一推式子:

$$\begin{aligned} A(x)B'(x) &\equiv 1 \pmod{x^{\lceil n/2 \rceil}} \\ (A(x)B'(x)-1)^2 &\equiv 0 \pmod{x^n} \\ A(x)^2B'(x)^2 - 2A(x)B'(x) + 1 &\equiv 0 \pmod{x^n} \\ 2A(x)B'(x) - A(x)^2B'(x)^2 &\equiv 1 \pmod{x^n} \\ A(x)\left(2B'(x) - A(x)B'(x)^2\right) &\equiv 1 \pmod{x^n} \end{aligned}$$

所以我们有

$$B(x) = 2B'(x) - A(x)B'(x)^2$$

特别注意这里带入的 $A(x)$ 应当是一个 n 次多项式而非 $\lceil n/2 \rceil$ 次多项式, 因为最后转化成的是 $A(x)B(x) \equiv 1$ 的形式, 如果 $A(x)$ 只有 $\lceil n/2 \rceil$ 项就不对了.

时间复杂度

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

多项式开方

问题: 对于给定的 $A(x)$, 求出 $B(x)$, 满足:

$$B(x)^2 \equiv A(x) \pmod{x^n}$$

跟多项式求逆一样的套路.
假设我们已经求出了 $B'(x)^2 \equiv A(x) \pmod{x^{\lceil n/2 \rceil}}$.
推一推式子:

$$
\begin{aligned}
B'(x)^2 &\equiv A(x) & \pmod{x^{\lceil n/2 \rceil}} \\
\left(B'(x)^2 - A(x)\right)^2 &\equiv 0 & \pmod{x^n} \\
\left(B'(x)^2 + A(x)\right)^2 &\equiv 4B'(x)^2 A(x) & \pmod{x^n} \\
\left(\frac{B'(x)^2 + A(x)}{2B'(x)}\right)^2 &\equiv A(x) & \pmod{x^n} \\
\left[\frac{1}{2}\left(B'(x) + A(x)B'(x)^{-1}\right)\right]^2 &\equiv A(x) & \pmod{x^n}
\end{aligned}
$$

仍然需要注意各项的次数.
$A(x)$ 应当是 n 次的, 需要注意的是 $B'(x)^{-1}$ 是 $B'(x)$ 在 $\pmod{x^n}$ 下的逆元.
时间复杂度也是 $O(n \log n)$.

其他

- 多项式除法
- 牛顿迭代 (多项式对数, 多项式指数)
- ...

啊好大.

由于时间有限, 有兴趣的同学自己去了解一下即可.
其实口胡并不难?

吐嘈

这些东西有什么卵用?

Codeforces 438E, The Child and Binary Tree

给你 n 个权值 $c_1, c_2, ..., c_n$. 同时定义一棵带权二叉树的权值为每个点的权值和.

对于每一个 $s \in [1, m]$, 请你求出权值为 s 的二叉树的个数. 注意二叉树的左右子树视作是不同的.

对 998244353 取模.

$n, m, c_i \leq 10^5$

这是一场 Chinese round...
首先考虑 DP.
显然有

$$dp(s) = \sum_{i=1}^{n} \sum_{c_i+j+k=s} dp(j)dp(k)$$

发现 $c_i + j + k = s$ 是一个卷积的形式, 考虑生成函数.

设 $C(x)$ 为权值的生成函数 (即 $[x^{c_i}]C = 1$), 同时 $F(x)$ 为 dp 的生成函数.
显然有

$$F(x) \equiv C(x)F(x)^2 + 1 \pmod{x^m}$$

解方程得到

$$F(x) \equiv \frac{1 \pm \sqrt{1 - 4C(x)}}{2C(x)} \pmod{x^m} \quad (1)$$

$$\equiv \frac{2}{1 \mp \sqrt{1 - 4C(x)}} \pmod{x^m} \quad (2)$$

注意 $[x^0]\sqrt{F(x)} > 0$.

为什么推到 (1) 之后还要变为 (2) 呢?
因为 $[x^0]C = 0$, 所以 $C(x)$ 在 $\pmod{x^m}$ 意义下并不存在逆元.
但是 $1 \mp \sqrt{1-4C(x)}$ 一定就能取到逆元?

稍微观察一下可以发现, 取 $+$ 的时候一定有逆元 ($[x^0](1+\sqrt{1-4C(x)}) = 2$), 而取 $-$ 的时候一定没有逆元 ($[x^0](1-\sqrt{1-4C(x)}) = 0$).
所以最终我们有

$$F(x) \equiv \frac{2}{1+\sqrt{1-4C(x)}} \pmod{x^m}$$

多项式开个方, 求个逆就可以了.
时间复杂度 $O(n + m\log m)$.

BZOJ 3456, 城市规划

请求出有 N 个点的有标号简单连通无向图的个数.
对 1004535809 取模.

$N \le 1.3 \times 10^5$

这题难在要求图连通. 不妨用类似反演的思想.

设 $f(n)$ 表示 n 个点时的答案, $g(n)$ 表示不要求一定连通时的方案, 显然 $g(n) = 2^{\binom{n}{2}}$.

枚举 1 号点所在的连通分量的大小, 则有

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

设

$$F(x)=\sum_{i=1}^{n}\frac{f(i)}{(i-1)!}x^i$$
$$G(x)=\sum_{i=1}^{n}\frac{g(i)}{i!}x^i$$
$$H(x)=\sum_{i=1}^{n}\frac{g(i)}{(i-1)!}x^i$$

则显然有

$$H(x)=F(x)G(x)$$

即

$$F(x)\equiv H(x)G(x)^{-1}\pmod{x^{N+1}}$$

然后多项式求逆就好了.

其实这题还有一种做法...
分治 FFT!

分治 FFT

给你一个递推式

$$f(n)=\sum_{i=0}^{n-1} f(i)a(n-i)$$

现在要求 $f(1)...f(n)$.
其中 $f(0)$ 和 $a(x)$ 都是已知的.

考虑 CDQ 分治.

现在要求 $f(l)...f(r-1)$.
设 $mid=\frac{l+r}{2}$, 则先递归求 $f(l)...f(mid-1)$, 再考虑 $[l,mid)$ 对 $[mid,r)$ 的贡献.
实际上只需把 $a(0)...a(r-l-1)$ 和 $f(l)...f(mid-1)$ 这两个卷起来就可以了, 取 $(mid-l)...(r-l-1)$ 项加入 $f(mid)...f(r-1)$ 即可.

机子太慢?

这里有一个卡常技巧.

令 $l_1 = r - l - 1$, $l_2 = mid - l$.
如果直接做, 我们需要做的长度是 $l_1 + l_2 - 1$ 的.
但是实际上我们只要做 l_1 长度的 FFT 就够了.

因为 FFT 可以理解为对一群单位根的一个快速插值算法.
当答案为一个 n 次多项式, 但是我们用 $r(r \le k)$ 次单位根给它插值的时候, 得到的 $r - 1$ 次多项式与本应得到的 n 次多项式相比, n 次多项式中 x_j 的系数会加到 $x^{j \bmod r}$ 的系数中去. (注意 $\omega_r^r = 1$)

然后不难验证, 对于上面的问题, 如果我们只做长度为 l_1 的 FFT, 那么我们所需的最后 l_2 项并不会被**循环卷积**影响到.

回到那道题吧...

直接上式子好了, 跟之前的思路是一样的

$$f(n) = 2^{\binom{n}{2}} - \sum_{i=1}^{n-1} f(i) 2^{\binom{n-i}{2}} \binom{n-1}{i-1}$$

$$\frac{f(n)}{(n-1)!} = \frac{2^{\binom{n}{2}}}{(n-1)!} - \sum_{i=1}^{n-1} \frac{f(i)}{(i-1)!} \frac{2^{\binom{n-i}{2}}}{(n-i)!}$$

然后直接分治 FFT 就行了.
时间复杂度 $O(n \log^2 n)$.

讲完了?

怎么没看见指数生成函数?

有 c 种颜色的巧克力, 每种颜色有无限个. 现在每次取出一个巧克力, 其颜色等概率为 $1...c$ 中的一种.
问最终有 m 种颜色的巧克力个数为奇数的概率.
对 998244353 取模.

$n \le 10^9, m \le c \le 10^5$

首先将问题转化为一个计数问题:

一个长度为 n 的数列, 每个位置是 $1...c$ 中任一个数, 求有多少种方案使得出现次数为奇数的值为 m 个.

排列问题, 考虑指数生成函数.

- 出现次数为奇数的颜色的生成函数:

$$f(x)=\frac{e^x-e^{-x}}{2}=x+\frac{x^3}{3!}+\frac{x^5}{5!}+\cdots$$

- 出现次数为偶数的颜色的生成函数:

$$g(x)=\frac{e^x+e^{-x}}{2}=1+\frac{x^2}{2!}+\frac{x^4}{4!}+\cdots$$

然后方案数就是

$$\binom{c}{m}\times n!\times[x^n]\left(f(x)^m g(x)^{c-m}\right)$$

但是由于 n 非常大, 这样搞显然不行. (况且 $n!$ 求得出来?)
考虑一种常见思路: 求出 e^x 为变量的多项式, 然后考虑每个 e^{kx} 对 x^n 的贡献, 也就是 $\frac{k^n}{n!}$.
发现同时 $n!$ 被消掉了!

然后直接 FFT 就好了.
注意由于这里不是在模意义下的多项式快速幂, 所以可以不用分治, 直接对点值快速幂即可.
时间复杂度 $O(c\log c)$.

完结撒花

这些都是多项式最为基础的内容.
其实无论是多项式的基本运算，还是一些题目的模型转化，都可以非常困难.
就先讲这么多吧.

Thank you!

~~Hope you slept comfortably!~~

References

1. Owaski, “多项式相关”
2. Picks’s Blog , “Inverse Element of Polynomial”
3. Picks’s Blog , “Square Root of Polynomial”
4. jiry_2’s Blog, “论逗逼的自我修养之 FFT 练习记”
5. New Meta’s Wiki, “分治 fft 的正确姿势”
6. zhangzj, “HNOI2016 模拟题 solution”