多项式与组合数学

2022.6.23

前置知识：泰勒展开

因此我们有了泰勒公式：

$$f(x)=f(x_0)+\sum_{i=1}^{n}\frac{f^{(i)}(x_0)}{i!}(x-x_0)^i+R_n(x)$$

当$x_0=0$的时候这就是麦克劳林公式：

$$f(x)=f(0)+\sum_{i=1}^{n}\frac{f^{(i)}(0)}{i!}x^i+R_n(x)$$

- 推导的方式很多，这里略过
- 常见的可以无限次求导的函数基本上都满足在$n\to\infty$时$R_{n(x)}\to 0$

定义，约定，符号

- 多项式：
  - 这里为了方便，我们所说的多项式（“若干个单项式的和”）也包括单项式（“数字和字母的积，或者单独的数字”）；
  - 我们今天要考虑的多项式，字母均只有一种，均用x代表
  - 将组成多项式的单项式中x的次数由低到高排序，一个n次多项式可以表示为$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$的形式。在设计程序时，我们往往用一个数组$\{a_0, a_1, a_2, \cdots a_n\}$来记录一个多项式的有关信息
  - $[x^n]f(x)$表示$f(x)$中x^n的系数，即前述的a_n
- 累加与累乘
  - $\sum_{i=l}^{r} f(i) = f(l) + f(l+1) + \cdots f(r-1) + f(r)$
  - $\prod_{i=l}^{r} f(i) = f(l) \times f(l+1) \times \cdots f(r-1) \times f(r)$

定义，约定，符号

- 关于e^X
  - 根据泰勒展开，我们有 $e^X \approx 1 + \frac{X^1}{1!} + \frac{X^2}{2!} + \cdots \frac{X^n}{n!} \cdots$，并且当右边的项数为无穷多时，可以认为余项（左右两边的差）无穷小，也就是左右两边相等
  - e^X中的X可以是一个数，一个字母，或者一个多项式（$e^{f(x)}$是$f(x)$和e^x的复合函数，也等价于$f(x)$和e^x的泰勒展开式的复合函数）
- 同样地，由泰勒展开得：$\ln(1-x) = -\sum_{i=1}^{n} \frac{x^i}{n}$

补充：形式幂级数及运算

- 我们在组合数学中使用的生成函数，和我们高中数学（大一微积分）课上学过的实数函数，其实有相当大的差异：
  - 处理生成函数时，我们通常都不关心x的取值；但是我们写出的生成函数，对很多的x的取值都是不收敛的（没有意义的）
  - 生成函数中，定义e^x完全没有用到幂运算，而是直接用泰勒级数定义；$\ln x$也是直接用泰勒级数定义
  - ……
- 在生成函数的运用中，我们不认为它是一个“函数”，而认为是一种“形式”，即由系数序列$\{a_n\}$定义的形式幂级数（简称幂级数）
- 当我们用前述的方法定义幂级数的加法、减法、乘法（同多项式），求导、不定积分（用极限定义），复合，指数、对数（泰勒展开）运算时，可以证明，我们之前学过的使用于实数函数的法则仍然适用于形式幂级数

补充：形式幂级数及运算

- 也就是说，对于形式幂级数，仍然有以下规律成立：
  - 加法交换律、结合律，乘法交换律、结合律，乘法分配律
  - 导数公式，不定积分公式，以及微积分基本定理（求导和不定积分为“互逆”的）
  - 复合函数的求导法则，分部积分，换元积分
  - 函数复合运算的结合律
  - 指数和对数互为反函数，即$\ln\left(1+(\exp x-1)\right)=x$
  - $\exp(A+B)=\exp(A)\times\exp(B)$
  - ……
- 基本上，Oier需要接触的所有的实数函数的运算法则，对形式幂级数都仍然成立
- 感兴趣的同学可以参考rqy：浅谈 OI 中常用的一些生成函数运算的合法与正确性

多项式的常见运算及其组合意义

多项式乘法

- 例1：香蕉一次可以取1个，最多取4次，最少取2次；苹果一次可以取5个，最多取4次，最少一次都不取。要取n个水果，问有几种取法

- 答案是$[x^n]F(x)$，其中$F(x) = (x^2 + x^3 + x^4)(1 + x^5 + x^{10} + x^{15} + x^{20})$
  - 记$h(x) = (x^2 + x^3 + x^4), p(x) = (1 + x^5 + x^{10} + x^{15} + x^{20})$，其各项系数$\{h_i\} = \{0,0,1,1,1,0,0 \cdots\}, \{p_i\} = \{1,0,0,0,0,1,0,0,0,0,1, \cdots\}$
  - 由多项式乘法的定义可以得到：$[x^n]F(x) = \sum_{i+j=n} h_i \times p_j$

多项式乘法

- 例2：（背包问题）设$F_j(x)=\sum_i a_{j,i}x^i$，其中$a_{j,i}$表示对于前j种水果，总共选了i个，总共的方案数；设$G_j(x)=\sum_i c_{j,i}x^i$其中$c_{j,i}$表示第j种水果选i个的方案数（在前面的两个问题中，$c_{j,i}\in\{0,1\}$）
- 那么根据基本的组合数学（或者是dp）：$a_{j,i}=\sum_{k\le i} a_{j-1,k}\times c_{j,i-k}$
- 也就是$F_j(x)=F_{j-1}(x)\times G_j(x)$

多项式乘法

- 例3：与上一道题相同，但是香蕉最少取一次，最多取无数次。即：香蕉一次可以取1个，最少取一次，最多取无数次；苹果一次可以取5个，最多取4次，最少一次都不取。要取n个水果，问有几种取法
- 同理可得答案$[x^n]F(x)$，其中$F(x)=(1+x+x^2+x^3+\cdots)(1+x^5+x^{10}+x^{15}+x^{20})$
- 处理方法：我们既然要计算x^n的系数，那么$(1+x+x^2+\cdots)$中次数高于n的项，与另一个括号中的项相乘和次数仍然高于n，因此对x^n的系数不会有任何影响
- 因此答案又等于$[x^n](1+x+x^2+\cdots+x^n)(1+x^5+x^{10}+x^{15}+x^{20})$
- “我们既然要计算x^n的系数，那么$(1+x+x^2+\cdots)$中次数高于n的项，与另一个括号中的项相乘后次数仍然高于n，因此对x^n的系数不会有任何影响”
- 在使用多项式时，我们要求的东西常常是一个多项式的前n项或者第n项，因此常常只需要算最低的n项的系数。只保留$F(x)$的最低的n项（$1,x,x^2,\cdots x^{n-1}$），令高次项系数全部为0，得到$F_1(x)$，记作$F_1(x)=F(x)(mod\ x^n)$

多项式的乘法逆

- 例3：设$g(x) = 1 + x + x^2 + \cdots$，即例3中的那个有无穷多项的多项式。根据等比数列求和可知$g(x) = \frac{1}{1-x}$，这个式子有什么含义呢？
- 观察发现：$g(x) \times (1 - x) = 1$
- 这给了我们求$g(x)$的另一种思路：我们只要求出一个与$(1 - x)$相乘后等于1的多项式就可以了；通常我们只需要$g(x)$的最低的n项的系数
- 因此我们这样定义多项式的乘法逆：对于某一多项式$f(x)$，如果存在最高次项次数不超过$n - 1$的多项式$g(x)$，使得$f(x) \times g(x) \equiv 1 (mod\ x^n)$成立，那么就称$g(x)$为$f(x)$在模$x^n$意义下的乘法逆
- 绝大多数时候我们说多项式求逆，指的就是求乘法逆

多项式的乘法逆

- 可以证明，多项式的乘法逆当且仅当$a_0 \neq 0$时存在，并当它存在时，它一定是唯一的（可以参考rqy：浅谈 OI 中常用的一些生成函数运算的合法与正确性）
- 多项式求逆可以在$\Theta(n \log n)$的复杂度实现

和卷积，差卷积

- 构造多项式以在Θ(n log n)的复杂度求：

$$c_i = \sum_{j+k=i} b_j \times c_k$$

$$c_i = \sum_{j-k=i} b_j \times c_k$$

多项式除法（取余）

- 这个除法是带余除法，也就是初中学因式分解的时候用过的大除法（一定要与乘法逆区分开！）
- 举个例子：$(x^3+3x^2+3x+5)\div(x^2+x+3)=(x+2)\cdots\cdots(-2x-1)$
- 可以在$\Theta(n\log n)$的复杂度实现，其中n是被除式的项数
- 常见应用是做常系数齐次线性递推

多项式的EXP，LN

- 有算法可以在$\Theta(n\log n)$的复杂度下求出$e^{f(x)}\ mod\ x^n$和$\ln f(x)\ \ mod\ x^n$

牛顿迭代

问题：已知$F(x)$，且$F(G(x))=0 \pmod{x^n}$，求$G(x)$。

考虑倍增，设已经求出了满足$F(G_i(x))=0 \pmod{x^{2^i}}$的$G_i(x)$。将$F(G_{i+1}(x))$在$G_i(x)$处泰勒展开，得：

$$F(G_{i+1}(x))=\sum_{i=0}^{\infty}\frac{F^{(i)}(G_i(x))}{i!}(G_{i+1}-G_i)^i$$

$$=F(G_i(x))+F'(G_i(x))[G_{i+1}(x)-G_i(x)]=0 \pmod{x^{2^{i+1}}}$$

整理得

$$G_{i+1}(x)=G_i(x)-\frac{F(G_i(x))}{F'(G_i(x))}$$

！注意：应用的前提时，在模x^k意义下$G(x)$是唯一的

一些实现

- 一种多项式求逆的好记的推导：令$F\big(G(x)\big)=\frac{1}{G(x)}-A(x)$，其中$A(x)$为需要被求逆的多项式，简单化简之后得到$G_{i+1}(x)=2G_i(x)-A(x)G_i^2(x)$
- 多项式ln：$\ln A(x)=\int\frac{A'(x)}{A(x)}dx$
- 多项式exp：令$F\big(G(x)\big)=\ln G(x)-A(x)$
- 多项式快速幂：允许取模时，ln + exp；不允许取模时，直接快速幂+多项式乘法

一些实现

多项式带余除法

带入$x=\frac{1}{x}$得

$$F(\frac{1}{x})=G(\frac{1}{x})Q(\frac{1}{x})+R(\frac{1}{x})$$

两边同时乘上x^n:

$$\begin{aligned} F^R(x) &= G^R(x)Q^R(x)+x^{n-m}R^R(x) \\ F^R(x) &= G^R(x)Q^R(x) \pmod{x^{n-m}} \end{aligned}$$

对$G^R(x)$求逆即可得到$Q(x)$。

一些实现

- 一种多项式求逆的推导：$G_i(x)-G_{i+1}(x)\equiv 0 \ mod \ x^{2^n} \Longrightarrow G_i^2(x)-2G_i(x)G_{i+1}(x)+G_{i+1}^2(x)\equiv 0 \ mod \ x^{2^{n+1}} \Longrightarrow A(x)G_i^2(x)-2G_{i+1}(x)+G_{i+1}(x)\equiv 0 \ mod \ x^{2^{n+1}}$

拉格朗日插值

有n个点$(x_1,y_1),(x_2,y_2)\cdots(x_n,y_n)$，满足$\forall i\neq j,x_i\neq x_j$。你需要求出一个次数小于等于$n-1$的多项式$f(x)$，使得$\forall i\in[1,n],f(x_i)=y_i$。

考虑对于每一个i，构造出一个在x_i处取值为1、在另外$n-1$个点取值为0的多项式$\ell_i(x)$：

$$\ell_i(x)=\frac{\prod_{j\neq i}(x-x_j)}{\prod_{j\neq i}(x_i-x_j)}$$

这个多项式显然是满足上面的条件的。

由此我们可以构造出一个多项式$L(x)=\sum_{i=1}^{n}y_i\ell_i(x)$。这个多项式显然满足在$x_i$点取值为$y_i$。

分治FFT

- 给定$\{g_1, g_2 \cdots g_n\}$，已知$f_n = \sum_{i=1}^{n} f_{n-i} g_i \ (n > 0), f_0 = 1$。求$\{f_1, f_2, \cdots f_n\}$。其中$n \leq 10^5$

多项式与组合数学中的经典问题

拉格朗日插值

- bzoj#3453. tyvj 1858 XLkxc

给定 k,a,n,d,p

f(i)=1^k+2^k+3^k+......+i^k

g(x)=f(1)+f(2)+f(3)+....+f(x)

求(g(a)+g(a+d)+g(a+2d)+......+g(a+nd))mod p

对于所有数据

1<=k<=123

0<=a,n,d<=123456789

p==1234567891

拉格朗日插值

- 下证明，答案是关于n的$k+3$次多项式
- 首先证明，当$n>1,k>1$，$f(n)=\sum_{i=1}^{n} i^k$是关于n的$k+1$次多项式

$$\sum_{i=1}^{n} i^k = \sum_{i=1}^{n} \sum_{j=1}^{\min(i,k)} \binom{i}{j} \left\{ {k \atop j} \right\} j!$$

$$= \sum_{j=1}^{k} \left\{ {k \atop j} \right\} j! \sum_{i=j}^{n} \binom{i}{j} = \sum_{j=1}^{k} \left\{ {k \atop j} \right\} j! \binom{n+1}{j+1}$$

拉格朗日插值

- Lemma：$f(n)$是关于n的k次多项式，则$g(n)=f(1)+f(2)+f(3)+\cdots f(n)$是关于$n$的 $k+1$次多项式
- $\sum_{i=1}^{n} f(n)=\sum_{j=0}^{k} a_j \sum_{i=1}^{n} i^j=\sum_{j=0}^{k} a_j \cdot$（$j+1$次多项式） ＝ （$k+1$次多项式）

- 于是可以得到$g(n)$是关于n的$k+2$次多项式
- 那么$g(an+d)$是关于n的$k+2$次多项式，答案则是关于n的$k+3$次多项式

拉格朗日插值

- 假设我们已经知道了$h(n)$是关于n的k次多项式，并且可以均摊$O(1)$地算出$h(1), h(2), \cdots h(k+4)$这$k+1$个点值，现在想要对某个$n$算出$h(n)$
- 答案即$\sum_{j=1}^{k+1} h(j) \cdot \prod_{1 \le i \le k+1, i \ne j} \frac{n-j}{i-j}$
- 提前处理好阶乘及其逆元，时间复杂度$O(k)$

拉格朗日插值

- 对于某个n，$g(an+d)$可以$O(k)$算出
- 于是可以在$O(k^2)$的时间算出$n=1,2,3\cdots k+4$的g值，再插值就可以得到最终的答案

二项式反演

- 本质上是容斥原理的一种特殊情况

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

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

二项式反演-证明

- 第一个公式的右侧，我们把$f(n)$换掉：$g(n)=\sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}\sum_{j=0}^{i}\binom{i}{j}g(j)$
- 交换$\sum$：$g(n)=\sum_{j=0}^{n}g(j)\sum_{i=j}^{n}(-1)^{n-i}\binom{n}{i}\binom{i}{j}$
- 只需证$\sum_{i=j}^{n}(-1)^{n-i}\binom{n}{i}\binom{i}{j}=[j=n]$
- 左侧即$\sum_{i=j}^{n}(-1)^{n-i}\binom{n}{j}\binom{n-j}{i-j}=(-1)^{n-j}\binom{n}{j}\sum_{k=0}^{n-j}(-1)^{k}\binom{n-j}{k}=(-1)^{n-j}\binom{n}{j}(1-1)^{n-j}=[j=n]$
- 证毕。第二个公式证明方法类似。

BZOJ2839 集合计数

- 题目：输入给出N，K。一个有N个元素的集合有2^N个不同子集（包含空集），现在要在这2^N个集合中取出若干集合（至少一个），使得它们的交集的元素个数为K，求取法的方案数，答案模1000000007。K<=N <= 1000000
- 设答案为$f(k)$，设钦定某i个元素必（不是所谓的“最少i个”）须在最终交集中，可能的方案数为$g(i)$
- 由题意有$g(k) = \sum_{i \geq k} \binom{i}{k} f(i)$，以及$g(k) = \binom{n}{k}\left(2^{2^{n-k}} - 1\right)$
- 二项式反演即可得到答案$f(k) = \sum_{i \geq k} (-1)^{i-k} \binom{i}{k} g(i)$
- 时空复杂度$\Theta(n)$

HDU6036 DIVISION GAME

桌上有k堆石头（编号从0到$k-1$），初始每一堆都有n个。第i轮会对第$(i-1) \mod k$堆石头进行操作。操作定义如下：从这一堆中拿走若干颗石头，使得拿走后，堆中石头的数量为原来这一堆石头的数量的因数。至少要拿走一个石头。当某一堆的石头数量为1时，游戏结束。问对于每个$i \in [0,k)$，第i堆石头为游戏结束前最后一堆被操作的石头的方案数。如果n的质因数分解为$n=\prod_{i=1}^{m} p_i^{e_i}$，那么将在输入中给出$m,(p_i,e_i)$。满足$p_i \le 10^9, \sum e_i \le 10^5, 1 \le m,k \le 10$。

- 一个常用的转化：$A=\prod_i p_i^{a_i}$是$B=\prod_i p_i^{b_i}$的因数，当且仅当$\forall i, a_i \le b_i$
- 我们对每个$j \in \sum e_i, l \in [0,k)$，算出仅考虑第$l$堆石头时，第$l$堆石头被操作了$j$次后个数变成1的方案数$s_{j,l}$；注意到操作$j$次后变成1的方案数等于操作$j-1$次后不变成1的方案数；这样就可以得到最终的答案

HDU6036 DIVISION GAME

- 如果没有“每次至少拿一个石头”这个限制，方案数是非常容易算的：$s'_{j,l} = \prod_i \binom{j+c_i-1}{j-1}$
- 有这个限制，就钦定若干步拿了0个石头，进行容斥反演，写出式子之后用多项式加速即可
- 计算的式子：$s_{j,l} = \sum_{y\le j}(-1)^y C_j^y s'_{y,l}$ 最终的复杂度是$\Theta(E\log E + kE)$，其中$E = \sum e_i$

第二类斯特林数的一行-拆组合数

- 问题：在$O(n\log n)$的复杂度中，求出$\{S_{n,1},S_{n,2},\cdots S_{n,n}\}$，其中$S_{n,k}$表示将$n$个有区别的元素划分成$k$个无区别的非空集合的方案数
- 常用的容斥公式$S_{n,k}=\frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n$（或者：$f(k)=k^n=\sum_{j\le k}\binom{k}{j}(j!\times S_{n,j})$
- 常用套路之拆组合数：$S_{n,k}=\frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\left(\frac{k!}{j!(k-j)!}\right)j^n=\frac{1}{k!}(-1)^k k!\sum_{j=0}^{k}\left(\frac{(-1)^{-j}}{j!}j^n\right)\left(\frac{1}{(k-j)!}\right)$
- 令$a_j=\left(\frac{(-1)^{-j}}{j!}j^n\right),b_j=\frac{1}{j!}$，则$S_{n,k}\cdot(-1)^{-k}=\sum_{j=0}^{k}a_j\cdot b_{k-j}$，就是前面的多项式乘法（$\{a_i\},\{b_i\}$看作多项式系数），可以用FFT快速解决

拆组合数？EGF！

- 如果$\{a_n\},\{b_n\},\{c_n\}$满足

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

- 我们常常把式子改写成

$$\frac{c_n}{n!}=\sum_{i=0}^{n}\frac{a_i}{i!}\times\frac{b_{n-i}}{(n-i)!}$$

- 从而可以用前面的说的多项式乘法来解决
- 更一般地，我们定义一个序列的普通生成函数（OGF）为$\sum a_ix^i$，定义它的指数生成函数（EGF）为$\sum a_i\frac{x^i}{i!}$

多项式指对运算的应用

- （BZOJ3456 城市规划）求出n个点的简单（无重边无自环）无向连通图数目，答案对1004535809取模。
- 如果不要求连通，方案数就是$f_n = 2^{C_n^2}$
- 设要求连通的方案数为g_n。设$F(x) = \sum_i \frac{f_i x^i}{i!}, G(x) = \sum_i \frac{g_i x^i}{i!}$，观察$F(x)$与$G(x)$之间的关系
- 发现$e^{G(x)} = \sum_{i \geq 0} \frac{\left(G(x)\right)^i}{i!} = F(x)$
- 求出$F(x)$之后，对$F(x)$取对即可。复杂度$\Theta(n \log n)$

CF438E THE CHILD AND BINARY TREE

- $n \le 10^5, m \le 10^5, 1 \le c_i \le 10^5$

我们的小朋友很喜欢计算机科学，而且尤其喜欢二叉树。 考虑一个含有 n 个互异正整数的序列 $c_1, c_2 \cdots, c_n$。如果一棵带点权的有根二叉树满足其所有顶点的权值都在集合 $\{c_1, c_2, \cdots, c_n\}$ 中，我们的小朋友就会将其称作神犇的。

并且他认为，一棵带点权的树的权值，是其所有顶点权值的总和。

给出一个整数 m，你能对于任意的 $1 \le s \le m$ 计算出权值为 s 的神犇二叉树的个数吗？请参照样例以更好的理解什么样的两棵二叉树会被视为不同的。 我们只需要知道答案关于 998244353 取模后的值。

Our child likes computer science very much, especially he likes binary trees.

Consider the sequence of n distinct positive integers: $c_1, c_2, \ldots, c_n$. The child calls a vertex-weighted rooted binary tree good if and only if for every vertex v , the weight of v is in the set $c_1, c_2, \ldots, c_n$. Also our child thinks that the weight of a vertex-weighted tree is the sum of all vertices' weights.

Given an integer m , can you for all s $(1 <= s <= m)$ calculate the number of good vertex-weighted rooted binary trees with weight s ? Please, check the samples for better understanding what trees are considered different.

We only want to know the answer modulo 998244353 ($7 \times 17 \times 2^{23} + 1$, a prime number).

CF438E THE CHILD AND BINARY TREE

- 设答案的普通生成函数$F(x) = 1 + \sum_{i=1}^{m} a_i x^i$，设$G(x) = \sum_{i=1}^{m} [i \in S]\, x^i$
- 则$F(x) = G(x)F^2(x) + 1$
- ~~牛顿迭代即可（？）~~
- 下一页ppt证明模x^k意义下满足方程的$F(x)$是唯一的
- 完成证明之后可以牛顿迭代，或者利用一元二次方程的求根公式，多项式开根求解

CF438E THE CHILD AND BINARY TREE

$$G(x)F^2(x) - F(x) + 1 = 0$$

$$\Longrightarrow\ F(x) = \frac{1 \pm \sqrt{1 - 4G(x)}}{2G(x)}$$

- 注意到$G(x)$常数项是0，但是$F(x)$又显然是存在的，所以分子的常数项一定也是0，分子分母同时约去x之后，分母会变成（常数项不为0）可以求逆的多项式
- 由朴素的（不用牛顿迭代的）多项式开根算法可知，$\sqrt{1 - 4G(x)}$存在且常数项为1，所以求根公式中只能取负号，从而这个方程在整式域上有唯一解
- 更简单的实现是$F(x) = \frac{2}{1+\sqrt{1-4G(x)}}$

递推式与网格路径

有一个$n \times n$的数组，其中，第一行和第一列的数字是给出的，而这个数组的其他位置的值满足下列递推式：

$$f[i,j] = a \times f[i,j-1] + b \times f[i-1,j] + c$$

求$f[n,n] \mod 10^6+3$的值。

- 数据范围：$n \leq 200000$ （source: bzoj4451 Frightful Formula）
- 设$g[1,i] = f[1,i], g[i,1] = f[i,1], g[i,j](i > 1, j > 1) = c$
- 将递推式“展开”，得到$Ans = \sum_{i=1}^{n}\sum_{j=1}^{n} g[i,j] \cdot a^{n-j} \cdot b^{n-i} C_{n-i+n-j}^{n-i}$
- 代码实现上，对每个$i+j$算出$\sum\left(\frac{a^{n-j}}{(n-j)!} \cdot \frac{b^{n-i}}{(n-i)!}\right)$即可；此题卡精度，用MTT可以通过

JZOJ6058 FALSE-FALSE-TRUE (FFT)

有$n+m$道题，其中有n道题的答案是yes，m道题的答案是no。小z并不知道哪些题是yes，哪些题是no，但是他知道n和m，并且在他答完一道题之后，他会知道自己是答对了还是答错了。输入n,m，问在小z采取最优策略的前提下，他答错的题目的数量期望。$n,m \le 500000$

- 这里只讲部分分：$n,m \le 10^5$；正解的做法（与这个部分分毫无关系）可以上网查题解

设$f_{i,j}$表示仍然剩下i道yes和j道no的时候，期望答错的题数。那么$f_{i,j} = \frac{i}{i+j} f_{i-1,j} + \frac{j}{i+j} f_{i,j-1} + \frac{\min(i,j)}{i+j}$。考虑每一个$(i,j)$的$\frac{\min(i,j)}{i+j}$对最终答案的贡献，可以推出：

$$Ans = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{\min(i,j)}{i+j} \frac{n!}{i!} \frac{m!}{j!} \frac{(i+j)!}{(n+m)!} \binom{n-i+m-j}{n-i}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{n!m!}{(n+m)!} \frac{(n+m-i-j)!(i+j)!}{(i+j)} \frac{1}{i!(n-i)!} \frac{1}{j!(m-j)!} \min(i,j)$$

JZOJ6058 FALSE-FALSE-TRUE (FFT)

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{n!m!}{(n+m)!}\frac{(n+m-i-j)!(i+j)!}{(i+j)}\frac{1}{i!(n-i)!}\frac{1}{j!(m-j)!}\min(i,j)$$

- 如果没有$\min(i,j)$，可以直接卷积
- 有$\min(i,j)$，套个分治就可以了
- 具体做法：假设当前要处理$i,j\in(l,r]$的部分，可以分别用卷积算出$i\in(l,mid],j\in(mid,r]$和$j\in(l,mid],i\in(mid,r]$的贡献，然后递归下去计算$i,j\in(l,mid]$和$i,j\in(mid,r]$的贡献
- 时间复杂度$\Theta(n\log^2 n)$

正睿NOI2019赛前集训D6T1 三角函数

求$\sum_{i=1}^{k} a_i \frac{\sin(x)}{x^i}$的$n$阶导数。

显然答案一定可以表示成$\sum_{i=1}^{n+k} b_i \frac{\sin(x)}{x^i} + \sum_{i=1}^{n+k} c_i \frac{\cos(x)}{x^i}$，你只需要求出$b_i$和$c_i$就可以了。

$n, k \le 10^5$，系数对998244353取模。

$$(\frac{\sin(x)}{x^i})' = \frac{\cos(x)x^i - \sin(x) \cdot ix^{i-1}}{x^{2i}} = \frac{\cos(x)}{x^i} - i\frac{\sin(x)}{x^{i+1}}$$

$$(\frac{\cos(x)}{x^i})' = \frac{-sin(x)x^i - \cos(x) \cdot ix^{i-1}}{x^{2i}} = -\frac{\sin(x)}{x^i} - i \cdot \frac{\cos(x)}{x^{i+1}}$$

- 尝试将递推式与路径联系！

正睿NOI2019赛前集训D6T1 三角函数

- 想象左边一列n个点，从上到下第i个点表示$\frac{\sin x}{x^i}$的系数，右边一列n个点，从上到下第i个点表示$\frac{\cos x}{x^i}$的系数
- 那么向右走$\times 1$，向左走$\times(-1)$，向下走$\times(-i)$，并且要求总步数为n
- 考虑每个a_i对答案的贡献，可以得到

$$b_j=\sum_{i<j,2|n-(j-i)} a_i\cdot\binom{n}{j-i}(-1)^{\frac{n-(j-i)}{2}}\cdot\frac{(-1)^{j-i}(j-1)!}{(i-1)!}$$

$$c_j=\sum_{i<j,2|n-(j-i)-1} a_i\cdot\binom{n}{j-i}(-1)^{\frac{n-(j-i)-1}{2}}\cdot\frac{(-1)^{j-i}(j-1)!}{(i-1)!}$$

- 时间复杂度$\Theta(n\log n)$

HDU5829 RIKKA WITH SUBSET

有一个包含n个数的集合$X=\{A_1,A_2\cdots A_n\}$。定义集合S对k的贡献$f(S,k)$为：S中前$min(k,|S|)$大的数（即当$|S|>k$时取前k大，否则取集合中的所有数）的和。
对于每个k，求$\sum_{Y\subset X} f(Y,k)$。$n\le 10^5, 0\le A_i\le 10^9$

答案对998244353取模

- 首先把答案差分一下，对每个k，求Y中第k大的元素的和
- 然后考虑每个元素对每个k的贡献
- 假设这个某个元素v_i是第i大，那么$Ans_k \mathrel{+}= v_i\cdot C_{i-1}^{k-1}\cdot 2^{n-i}$，即$\frac{Ans_k}{(k-1)!} \mathrel{+}= \left(\frac{v_i\cdot 2^{n-i}}{(i-1)!}\right)\left(\frac{1}{(k-i)!}\right)$
- NTT加速，复杂度$\Theta(n\log n)$

19十连D1T3

有一个无限大的二维数组F，它满足如下的限制：
如果$i<0$或者$j\le 0$，那么$F_{i,j}=0$。
如果$i=0$，那么$F_{i,j}=C_j$。
否则$F_{i,j}=F_{i-1,j}+F_{i,j-1}$。

给出$C_1,C_2,C_3,\cdots C_n$，有两种操作：1.将C_x修改为$y(x\le n)$。2.查询$F_{x,y}$的值对10^9+7取模的结果$(0\le x\le 20, 1\le y\le n)$。

数据范围$n\le 10^5$

19十连D1T3

- 沿用前面的思路，仍然考虑C_i对$F_{x,y}$的贡献
- 得到$F_{x,y}=\sum_{i\le y}C_i\cdot\binom{y-i+x}{x}$
- 这里有一个套路：由于x很小，我们可以在“处理输出$\{C_n\}$和修改操作”时枚举x，这以后可以将x视为常数，则$\binom{y-i+x}{x}$（视为下降幂除以$x!$）是一个次数不超过x的、关于y,i的多项式
- 假设多项式为$\sum_u\sum_v a_{u,v}y^u i^v$，其中$a_{u,v}$的取值可以预处理出来（不同的$x,u,v$只有20*20*20种）
- 可以在~~线段树~~树状数组中，对于每个$v\in[0,20]$，维护好一个区间$\sum C_i\cdot i^v$；查询的时候再算出上面多项式的值就可以了
- 复杂度$O(X^2Q+X(n+Q)\log n)$，其中X为询问中x的最大值

19十连D1T3

- 补充1：这道题也可以用类似《B君的第二题》的处理组合数的做法，需要用到负数的组合数，即 $C_{-a}^{b}=\frac{(-a)\cdot(-a-1)\cdots(-a-b+1)}{b!},(a,b>0)$
- 补充2：有一个非常简单的对询问分块的做法！存储下最近几次修改操作，并且在回答询问时，单独最近修改了的位置的影响，每经过了$\sqrt{Q}$次修改之后就重新预处理整个数组的信息，常数优秀的话是可以通过的

基础题（备选）

- HDU5885 XM Reserves（二维卷积）
- HDU5307 He is flying（拆i-j）
- JZOJ5702 [GDOI2018]滑稽子图（二项式定理）

进阶题

- AUOJ prob28 我的朋友们（分治FFT）
- BZOJ2498 Xavier is Learning to Count（容斥系数需要自己证明）
- P4389 付公主的背包（多项式指对运算的应用）
- JZOJ6054 Z的礼物（斯特林反演）