生成函数 & 多项式

zzq

生成函数是啥

- 对于一个数列$a_0, a_1, a_2 \ldots$，定义$f(x) = \sum_{i=0}^{n} a_i x^i$为它的普通生成函数（ordinary generating function，ogf)
- 对于一个数列$a_0, a_1, a_2 \ldots$，定义$f(x) = \sum_{i=0}^{n} a_i \frac{x^i}{i!}$为它的指数生成函数（exponential generating function，egf)
- 下文中未加说明，“生成函数”均指普通生成函数。

生成函数的运算

- 我们考察两个生成函数的乘积。
- 假设两个数列a,b分别对应两个普通生成函数f,g，那么令$h(x)=f(x)g(x)$，那么对应它的数列c就满足$c_t=\sum_{i=0}^{t}a_ib_{t-i}$，即普通生成函数的乘积就对应了数列的卷积。普通生成函数一般用于没有顺序的情况。
- 假设两个数列a,b分别对应两个指数生成函数f,g，那么令$h(x)=f(x)g(x)$，那么对应它的数列c就满足$c_t=t!\sum_{i=0}^{t}\frac{a_i}{i!}\frac{b_{t-i}}{(t-i)!}=\sum_{i=0}^{t}C_t^ia_ib_{t-i}$，即普通生成函数的乘积就对应了数列带组合数的卷积。例如我们有两种球有序地排成一列，第一种球有i个的方案数为a_i，第二种球有i个的方案数为b_i，那么排成一列共有t个球的方案数就满足$c_t=\sum_{i=0}^{t}C_t^ia_ib_{t-i}$。

生成函数的运算

- 加法就是加法，意义就是在两样东西中选一样，根据加法原理方案数要加起来。
- 指数生成函数的exp：记数列a的指数生成函数为f，考察e^f。
- 由泰勒展开，$e^f=\sum_{i=0}^{\infty}\frac{f^i}{i!}$，$f^i$的组合意义就是把i个不同的东西排列在一起，除掉$i!$就可以得到i个东西有序组合在一起的方案数（这i个东西本身的顺序无关紧要），所以它求出的就是用若干个f中的方案排出若干个东西的方案数。我们可以称exp求出的是f的无序组合。

生成函数的运算

- 一个经典的例子：
- 设f是无向连通图的指数生成函数，表示f。
- 设g是无向图的指数生成函数，i个点的无向图个数显然是$2^{C_i^2}$，那么我们有$\mathrm{g(x)} = \sum_{i=0}^{\infty} 2^{C_i^2} \frac{x^i}{i!}$。
- 这时我们注意到g就是f的无序组合，即我们先把点划分成若干个联通块，再把每个联通块搞成联通的，那么我们有$g = e^f$，所以$f = \ln(g)$。

生成函数有啥用

- 单靠生成函数就可以解决许多问题，这时生成函数的主要用途是推导出某些结论。~~被打表高手吊着打~~

生成函数有啥用

- 例1：已知卡特兰数C_n满足$C_0=1$，$C_{n+1}=\sum_{i=0}^{n} C_i C_{n-i}$，求出它的另类递推式（即将$C_n$用$n$和$C_{n-1}$表示）。

生成函数有啥用

- 例1：已知卡特兰数C_n满足$C_0=1$，$C_{n+1}=\sum_{i=0}^{n}C_iC_{n-i}$，求出它的另类递推式（即将$C_n$用$n$和$C_{n-1}$表示）。
- 考虑卡特兰数的普通生成函数f，那么容易看出$xf^2+1=f$。我们把它当做一个二次方程解一下，会解出$f=\frac{1\pm\sqrt{1-4x}}{2x}$，$\pm$是因为这里根号不是良定义的（类似$\sqrt{4}=\pm2$），本来就有正负两个解。这里我们设$\sqrt{1-4x}$的常数项为1，那么$f=\frac{1-\sqrt{1-4x}}{2x}$。
- 设$g=\sqrt{1-4x}$，g对应的数列为D。

生成函数有啥用

- 我们先证明一个小引理，设$F(x)=A(x)^p$，那么$A(x)F'(x)=pF(x)A'(x)$。
- 由幂法则，我们对$F(x)=A(x)^p$两侧求导，那么$F'(x)=pA(x)^{p-1}A'(x)$，把两侧乘上$A(x)$得到$F'(x)A(x)=pA(x)^pA'(x)$，又注意到$A(x)^p=F(x)$即证。
- 设$g=\sqrt{1-4x}$，即$g=(1-4x)^{0.5}$，那么$(1-4x)g'(x)=\frac{1}{2}g(x)(-4)=-2g(x)$。设其对应的数列为$D$，那么我们有$g(x)=\sum_{i=0}^{\infty}D_ix^i$，$g'(x)=\sum_{i=0}^{\infty}(i+1)D_{i+1}x^i$。带入原式，我们有$\sum_{i=0}^{\infty}(i+1)D_{i+1}x^i(1-4x)=-2\sum_{i=0}^{\infty}D_ix^i$。

生成函数有啥用

- $\sum_{i=0}^{\infty}(i+1)D_{i+1}x^i(1-4x) = -2\sum_{i=0}^{\infty}D_i x^i$
- $\sum_{i=0}^{\infty}(i+1)D_{i+1}x^i - 4\sum_{i=0}^{\infty}(i+1)D_{i+1}x^{i+1} = -2\sum_{i=0}^{\infty}D_i x^i$
- $\sum_{i=0}^{\infty}(i+1)D_{i+1}x^i - 4\sum_{i=1}^{\infty}iD_i x^i + 2\sum_{i=0}^{\infty}D_i x^i = 0$
- 比较$[x^i]$，我们有 $(i+1)D_{i+1} - 4iD_i + 2D_i = 0$ $(i \geq 1)$
- 整理得$\mathrm{D_n} = \frac{4n-6}{n}D_{n-1}$。

生成函数有啥用

- 由于$f=\frac{1-\sqrt{1-4x}}{2x}=\frac{1-g}{2x}$，那么同样对比系数可以发现$C_n=-\frac{D_{n+1}}{2}$，那么由于$\mathrm{D_n}=\frac{4n-6}{n}D_{n-1}$，$\mathrm{C_{n-1}}=\frac{4n-6}{n}\mathrm{C}_{n-2}$，即$\mathrm{C_n}=\frac{4n-2}{n+1}C_{n-1}$，大功告成。

生成函数有啥用

- 例2：对于一个排列，定义它的价值为它的环数的m次方，求所有1...n的排列的价值之和。对10^9+7取模。时限1s。
- $1 \le n \le 10^5$，$0 \le m \le 300$。

* Source: Topcoder CyclesNumber

生成函数有啥用

- 这题要用到一些斯特林数有关的知识。
- 第一类斯特林数：$s(n,k)$，表示将n个元素排列成k个圆排列的方案数
- 第二类斯特林数：$S(n,k)$，表示将n个元素划分成k个非空集合的方案数
- 斯特林数常用于一般幂和下降幂的转化，在第二部分我们还会再见到它们，这里先给出常用的结论。
- 记$x^{\overline{n}} = \prod_{i=0}^{n-1}(x+i)$，$x^{\underline{n}} = \prod_{i=0}^{n-1}(x-i)$，我们有$x^{\overline{n}} = \sum_{i=0}^{n} s(n,i)x^i$，$x^n = \sum_{i=0}^{n} S(n,i)x^{\underline{i}}$。证明可以利用组合意义，这里不赘述。

生成函数有啥用

- 题意就是求$\sum_{i=1}^{n} s(n,i)i^m$，那么我们考虑$s(n,i)$的普通生成函数$s_n(x)$（将n看作定值），那么$\mathrm{s_n(x)} = \sum_{i=0}^{n} s(n,i)x^i = x^{\overline{n}}$。
- i^m不太好办，我们把它写成下降幂：
- $\sum_{i=1}^{n} s(n,i)i^m = \sum_{i=1}^{n} s(n,i)\sum_{j=0}^{m} S(m,j)i^{\underline{j}} = \sum_{j=0}^{m} S(m,j)\sum_{i=1}^{n} s(n,i)i^{\underline{j}}$
- 注意到$(x^n)^{(j)} = x^{n-j}n^{\underline{j}}$（$n \ge j$，$f^{(j)}$指的是求f的j阶导），那么$\sum_{i=1}^{n} s(n,i)i^{\underline{j}}$就是$s_n(x)$求j阶导之后带入x=1得到的值。
- 带x=1不太方便，考虑设$g(x) = s_n(x+1)$。

生成函数有啥用

- $\mathrm{g(x)} = \mathrm{s_n(x+1)} = (x+1)^{\overline{n}} = \frac{x^{\overline{n+1}}}{x} = \frac{s_{n+1}(x)}{x} = \sum_{i=0}^{n} s(n+1, i+1)x^i$ 。
- 求j阶导之后带入x=0，那么就是要求求j阶导后的常数，即$s(n+1, j+1)j!$。带入原式就做完了。s和S可以$O(nm)$递推出来。

生成函数有啥用

- 例3：有排成一排的n个球，你要把每个球染成k种颜色之一，每种颜色的球的个数都要是d的倍数，问方案数mod 19491001。时限1s。
- $1 \le n \le 10^9$
- 有两个子任务：
  - $1 \le k \le 500000$，$\mathrm{d} = 2$
  - $1 \le k \le 1000$，$d = 3$

* Source: 集训队作业2019

生成函数有啥用

- 考虑每种颜色的指数生成函数：$\sum_{i=0}^{\infty}\frac{x^{id}}{(id)!}$，那么答案就是 $n!\,[x^n]\left(\sum_{i=0}^{\infty}\frac{x^{id}}{(id)!}\right)^k$。考虑化简$\sum_{i=0}^{\infty}\frac{x^{id}}{(id)!}$。
- 当$d=2$时，考虑$e^x=\sum_{i=0}^{\infty}\frac{x^i}{i!}$，我们只需要削掉所有奇次的项。
- 注意到$e^{-x}=\sum_{i=0}^{\infty}\frac{(-1)^i x^i}{i!}$，在奇次项处有-1的系数。
- 把两个式子相加，那么奇次项系数就是0了，偶次项系数是2。
- 所以$\sum_{i=0}^{\infty}\frac{x^{id}}{(id)!}=\frac{e^x+e^{-x}}{2}$。

生成函数有啥用

- 使用二项式定理展开$(e^x+e^{-x})^k$，那么会得到$\sum_{i=0}^{k} C_k^i e^{x(2i-k)}$，$[\mathrm{x}^{\mathrm{n}}]e^{kx}$就是$\frac{k^n}{n!}$，所以答案就是$\frac{1}{2^k}\sum_{i=0}^{k} C_k^i (2i-k)^n$，容易$O(k\log(n))$计算。
- 当d=3时，考虑推广这个思路。不难想到利用三次单位根ω_3，由于$1+\omega_3+\omega_3^2=0$，我们可以发现对于一个整数k，$\sum_{i=0}^{2} \omega_3^{ki}=3[3|k]$，即当k是3的倍数时为3，否则为0。
- 那么$\sum_{i=0}^{\infty}\frac{x^{id}}{(id)!}=\frac{e^x+e^{x\omega_3}+e^{x\omega_3^2}}{3}$。
- 注意到$19491001\equiv 1\ (mod\ 3)$，那么$3|\varphi(19491001)$，所以存在$mod\ 19491001$的三次单位根$g^{\frac{19491001-1}{3}}$（$g$为任意原根）。

生成函数有啥用

- 类似地，我们可以将$\left(\frac{e^{x}+e^{x\omega_3}+e^{x\omega_3^2}}{3}\right)^k$二项式展开，得到一个类似的式子，从而$O\left(k^2\log(n)\right)$进行计算。

多项式是啥

- 只用到生成函数的题就先介绍这么多了，在更多的题之前，我们需要多项式运算这一武器。
- 多项式是啥就不介绍了。

多项式基本运算

- 加法，$O(n)$。
- 减法，$O(n)$。
- 乘法，快速傅里叶变换，$O(n\log(n))$，这里假装大家都会。
  - $F(\omega_n^m) = F_0\left(\omega_{\frac{n}{2}}^m\right) + \omega_n^m F_1\left(\omega_{\frac{n}{2}}^m\right)$
  - $F\left(\omega_n^{m+\frac{n}{2}}\right) = F_0\left(\omega_{\frac{n}{2}}^m\right) - \omega_n^m F_1\left(\omega_{\frac{n}{2}}^m\right)$

多项式基本运算

- 牛顿迭代：
- 有一个关于多项式f的方程$g(f)=0$，其中g也是一个多项式，解出$f \bmod x^n$。
- 考虑每次倍增已知项的次数。假设我们已知$f \equiv f_0 \pmod{x^n}$，考虑算出$f \bmod x^{2n}$。
- 把g在f_0处泰勒展开，有$0=g(f)=g(f_0)+g'(f_0)(f-f_0)+\frac{g''(f_0)}{2}(f-f_0)^2+\cdots \equiv g(f_0)+g'(f_0)(f-f_0) \pmod{x^{2n}}$
- 所以$f \equiv f_0-\frac{g(f_0)}{g'(f_0)} \pmod{x^{2n}}$，倍增$n$即可，注意这只是一个框架，实际上$g(f_0)$和$g'(f_0)$并不容易计算。
- 推导时需要注意的是$g'(f_0)$是把f_0作为自变量，而不是x作为自变量。

多项式基本运算

- 利用牛顿迭代可以轻松推出很多基本运算。
- 求逆：有一个多项式f，求一个多项式g满足$fg \equiv 1 \pmod{x^n}$。
- 仍然倍增n，假设我们已知$g \equiv g_0 \pmod{x^n}$，那么我们带入牛迭式子，就有 $g \equiv g_0 - \frac{fg_0 - 1}{f} \pmod{x^{2n}}$
- 注意到$fg_0 \equiv 1 \pmod{x^n}$，那么这里的 $\frac{fg_0 - 1}{f} \equiv (fg_0 - 1)g_0 \pmod{x^{2n}}$。所以 $g \equiv g_0 - (fg_0 - 1)g_0 = 2g_0 - fg_0^2 \pmod{x^{2n}}$。
- $O(n\log(n))$

多项式基本运算

- 求ln：有一个多项式f，f的常数项为1，求$\ln(f) \pmod{x^n}$。
- 求这个需要一些数学知识：$\ln f(x)=\int \frac{f'(x)}{f(x)}dx$，除以$f(x)$可以改成乘逆元，求导和积分都是线性的。还是$O(n\log(n))$。
- 求exp：有一个多项式f，f的常数项为0，求 $g \equiv e^f \pmod{x^n}$。
- $g \equiv e^f$即$\ln(g) \equiv f$，同样倍增n，设$g \equiv g_0 \pmod{x^n}$，带入牛迭式子，我们有$g \equiv g_0-(\ln(g_0)-f)g_0 \pmod{x^{2n}}$。仍然是$O(n\log(n))$，不过常数大了很多。

多项式基本运算

- 快速幂：有一个多项式f和一个正整数t，求$f^t \pmod{x^n}$。
- $f^t \equiv e^{t\ln(f)}$，只需要一次ln和一次exp，$O(n\log(n))$。需要注意的是ln只能对常数为1的多项式使用，所以一开始要先去掉f尾部的0（最后补上0），然后把常数项改成1（最后乘上常数项的t次方）。
- 开方：有一个多项式f，求一个多项式g使$g^2 \equiv f \pmod{x^n}$。同样可以牛顿迭代，但是更简单的方法是直接用快速幂的方法求1/2次幂。

多项式基本运算

- 除法和取模：对于多项式A, B，求出多项式D, R，满足$A(x) = B(x)D(x) + R(x)$，$\deg D = \deg A - \deg B$，$\deg R \le \deg B - 1$，其中$\deg F$表示F最高此项的次数。

- 定义$n = \deg A$，$m = \deg B$，考虑设$F^R(x) = x^{\deg F} F\left(\frac{1}{x}\right)$，即$F$各项系数翻转的结果。

- 用$\frac{1}{x}$带入原式：$A\left(\frac{1}{x}\right) = B\left(\frac{1}{x}\right) D\left(\frac{1}{x}\right) + R(\frac{1}{x})$，两边乘上$x^n$：$\mathrm{x}^{\mathrm{n}} A\left(\frac{1}{x}\right) = x^m B\left(\frac{1}{x}\right) x^{n-m} D\left(\frac{1}{x}\right) + x^{n-m+1} x^{m-1} R(\frac{1}{x})$，即$A^R(x) = B^R(x)D^R(x) + x^{n-m+1} R^R(x)$，两边模$x^{n-m+1}$就有$D^R(x) \equiv A^R(x)[B^R(x)]^{-1} \pmod{x^{n-\mathrm{m}+1}}$。

- 这样求出D之后带回原式即可求出R。$O(n\log(n))$。

多项式基本运算

- 多点求值:
- 给一个多项式F和$x_1, x_2 \ldots x_n$，求出$F(x_1), F(x_2) \ldots F(x_n)$。
- 设$L(x) = \prod_{i=1}^{n/2}(x - x_i)$，$\mathrm{R(x)} = \prod_{i=\frac{n}{2}+1}^{n}(x - x_i)$。
- 那么对于$i \le \frac{n}{2}$，我们有$F(x_i) = (F \bmod L)(x_i)$，对于$i > \frac{n}{2}$，我们有$F(x_i) = (F \bmod R)(x_i)$，递归即可。
- 复杂度与分治fft相同，即$O(n\log^2(n))$。

多项式基本运算

- 多点插值：给$x_1, x_2 \dots x_n$，$y_1, y_2 \dots y_n$，你需要求出一个$n-1$次多项式F满足$F(x_i) = y_i$。
- 考虑拉格朗日插值，我们有$F(x) = \sum_{i=1}^{n} y_i \frac{\prod_{j \neq i}(x-x_j)}{\prod_{j \neq i}(x_i-x_j)}$。
- 先考虑对于每个i如何求出$\prod_{j \neq i}(x_i - x_j)$。我们设$M(x) = \prod_{i=1}^{n}(x - x_i)$，我们就要求$\frac{M(x)}{x-x_i}$。这个分数上下都为0，那么我们用洛必达法则，所以它就等于$M'(x_i)$。多点求值即可求出。
- 接下来记$\frac{y_i}{\prod_{j \neq i}(x_i-x_j)} = v_i$，我们就要求$\sum_{i=1}^{n} v_i \prod_{j \neq i}(x - x_j)$，这仍然可以分治fft。时间复杂度仍然是$O(n\log^2(n))$。

简单应用

- 例1，背包问题。
- 有一个大小为n的背包，你需要用一些物品把它装满。大小为i的物品共有a_i种，每种都有无穷个。求把背包装满的方案数，这里两个方案不同当且仅当某一种物品选取的数量不同。例如取$\mathrm{a}_i = 1$即为拆分数。
- 使用普通生成函数，那么答案就是$[\mathrm{x}^n]\prod_{i=1}^{n}\left(\frac{1}{1-x^i}\right)^{a_i}$。

简单应用

- $\prod_{i=1}^{n}\left(\frac{1}{1-x^i}\right)^{a_i} = \exp(-\sum_{i=1}^{n} a_i \ln(1-x^i)) = \exp\left(-\sum_{i=1}^{n} a_i \sum_{j\geq 1}\frac{x^{ij}}{j}\right) = \exp\left(\sum_{j\geq 1}\frac{1}{j}A(x^j)\right)$。
- 枚举j之后暴力求出$\sum_{j\geq 1}\frac{1}{j}A(x^j)$即可，复杂度显然是调和级数，最后求一次exp。仍然是$O(n\log(n))$。

简单应用

- 例2，小朋友和二叉树。
- 给一个集合S，对每个$x \in [1, m]$，问有多少个不同的带权二叉树满足其每个点的权值在S里面，且权值和=x，输出方案数mod 998244353。
- $n, m \leq 100000$。

* Source: Codeforces Round #250

简单应用

- 记$f(x)$为答案的生成函数，$g(x)$为S的普通生成函数，那么考虑左右孩子，我们有$f(x) = g(x)f^2(x) + 1$。
- 所以$f(x) = \frac{2}{1 \pm \sqrt{1-4g(x)}}$。$O(n\log(n))$。

简单应用

- 例3，Counting Sheep in Ami Dongsuo
- 有一张n个点m条边的有向无环图，每个点上写有一个不超过w的正整数。有三个人要从某个点出发，每个人任意选择一条路径走到某个点，把三个人走到的点点权相加就是这次旅行的特征值。求特征值为0...3w的旅行个数$\mathrm{mod}\ 10^9+7$。
- $1 \le n \le 10000$，$1 \le m \le 30000$，$1 \le w \le 400$，2s。

* Source: 2018 Qingdao Regional

简单应用

- 一种简单的想法是把每个点出发的每种点权结束的路径数全算出来，然后求3次幂输出，但是这样并不能过。
- 考虑插值，即把每个点点权改成一个x^w的系数，这样求3次幂就可以真的3次幂了，就相当于指数相加。带入$x = 0 \dots 3w$，把每个点的值加起来，最后拉格朗日插值插回去就可以得到答案了。

简单应用

- 例4，fuzzy search。
- 有一个大串$s_1, s_2 \dots s_n$和一个小串$t_1, t_2 \dots t_m$，你要找一个大串的长度为m的小串与s匹配，最小化它们之间的距离。两个串$a_1, a_2 \dots a_m$和$b_1, b_2 \dots b_m$的距离定义为$\sum_{i=1}^{m} |a_i - b_i|$。$s_i, t_i \in [1,4]$，长度$10^5$。

* Source: 原创

简单应用

- 如果距离是$a_i b_i$，对于某个匹配开头s我们就是要求$\sum_{i=1}^{m} a_{s+\mathrm{i}-1} b_i$，我们记$c_i = b_{m-i}$，那么就是要求$\sum_{i=1}^{m} a_{s+\mathrm{i}-1} b_{m-i}$。将$a$与$c$卷积，那么s+m-1次系数就是欲求的。
- 考虑把距离转化成这种形式。我们在$\mathrm{mod}\ 998244353$下做ntt的话，是存在8次单位根ω_8的（熟知8|998244352）。考虑直接求出每个位置每个差值各有多少个。把a_i改为$\omega_8^{a_i}$，b_i改为$\omega_8^{-b_i}$，然后把距离改成$a_i b_i$，那么差为$-3, -2 \ldots 3$分别对应$\omega_8^5, \omega_8^6 \ldots \omega_8^3$项的系数。
- 带入$x = \omega_8^i\ (i \in [0,7])$做8次ntt并插值出系数即可。

伯努利数

- 伯努利数是解决自然数幂和问题的重要工具。
- 我们定义伯努利数B满足$\frac{x}{e^x-1}=\sum_{n=0}^{\infty}\frac{B_n}{n!}x^n$。
- 定义伯努利数多项式$\beta_n(t)$满足：
$\sum_{n=0}^{\infty}\frac{\beta_n(t)}{n!}x^n=\frac{x}{e^x-1}e^{tx}=(\sum_{n=0}^{\infty}\frac{B_n}{n!}x^n)(\sum_{n=0}^{\infty}\frac{t^n}{n!}x^n)$。
- 那么$\beta_n(t)=\sum_{k=0}^{n}C_n^k B_{n-k}t^k$（即这两个egf相乘）
- $\sum_{n=0}^{\infty}\frac{\beta_n(t+1)-\beta_n(t)}{n!}x^n=\frac{x}{e^x-1}e^{tx}(e^x-1)=xe^{tx}=x\sum_{n=0}^{\infty}\frac{t^n}{n!}x^n$。
- 比较系数有$\beta_{n+1}(t+1)-\beta_{n+1}(t)=(n+1)t^n$，
所以$\sum_{i=0}^{n}i^d=\frac{\beta_{d+1}(n+1)-\beta_{d+1}(0)}{d+1}$。

伯努利数

- $\sum_{i=0}^{n} i^d = \frac{\beta_{d+1}(n+1)-\beta_{d+1}(0)}{d+1} = \frac{1}{d+1}\sum_{g=0}^{d} C_{d+1}^{g} B_g n^{d+1-g}$ 。
- 求伯努利数本身只需要对$e^x - 1$求逆即可，那么我们就可以方便地求出自然数幂和了。
- 当然如果只是求自然数幂和有更方便的方法，例如直接拉格朗日插值，但是伯努利数还是有一些优点的。

简单应用

- 例5，仓鼠的数学题。
- 设$\mathrm{S_k(x)}=\sum_{i=0}^{x} i^k$，输入一个数列$a_0, a_1 \ldots a_n$，求$\sum_{k=0}^{n} S_k(x) a_k$的各项系数 mod 998244353。

* Source: 原创

简单应用

- 用伯努利数的式子带入：$\sum_{k=0}^{n} S_k(x)a_k = \sum_{k=0}^{n} \frac{a_k}{k+1} \sum_{g=0}^{k} C_{k+1}^{g} B_g x^{k+1-g} = \sum_{k=0}^{n} a_k k! \sum_{g=0}^{n} \frac{B_g}{g!} \frac{x^{k+1-g}}{(k+1-g)!}$。
- 接下来的工作就很简单了，设$X_k = a_k k!$，$Y_k = \frac{B_k}{k!}$，将Y倒过来卷积即可。
- 如前所述，求伯努利数只需要一次求逆。$O(n\log(n))$。

简单应用

- 例6，生成n个随机数，第i个在$[0, t_i]$中均匀随机，求这些实数之和的m次方的期望。
- $1 \leq n, m \leq 10^5$，2s。

* Source: 原创

简单应用

- 设第i次随机出了x_i，那么我们就要求$E((\sum_i x_i)^m)$。
- 首先我们注意到x_i^m的期望为$\frac{r_i^m}{m+1}$，这是因为$\int_0^1 x^m\,dx = \frac{1}{m+1}$。
- 考虑$E\left((\sum_{i=1}^n x_i)^m\right) = \sum_{i=0}^m C_m^i E\left(x_n^i\right) E\left(\left(\sum_{i=1}^{n-1} x_i\right)^{m-i}\right)$，这就是一个指数生成函数的形式，我们设$F_t(x) = \sum_{i=0}^m E\left(x_t^i\right)\frac{x^i}{i!} = \sum_{i=0}^m \frac{r_t^i x^i}{(i+1)!}$，答案就是$m!\,[x^m]\prod_{i=1}^n F_i(x)$。
- 设$F(x) = \sum_{i=0}^m \frac{x^i}{(i+1)!}$，那么$F_t(x) = F(r_t x)$，$\prod_{i=1}^n F_i(x) = \exp\left(\sum_{i=1}^n \ln\left(F_i(x)\right)\right)$。

简单应用

- 我们求出$\ln(F(x)) = \sum_{i=0}^{m} p_i x^i$，那么$\sum_{i=1}^{n} \ln(F_i(x)) = \sum_{i=1}^{n} \sum_{j=0}^{m} p_j (xr_i)^j = \sum_{j=0}^{m} p_j x^j (\sum_{i=1}^{n} r_i^j)$
- 那么我们就只需要对于每个$j \in [0, m]$求出$\sum_{i=1}^{n} r_i^j$即可。
- 一种简单易懂的求法是注意到$\frac{1}{1-xr_i} = \sum_{j=0}^{\infty} x^j r_i^j$，所以我们只需要求出$\sum_{i=1}^{n} \frac{1}{1-x\mathrm{r}_i}$。
- 考虑进行暴力通分（即$\frac{a}{b} + \frac{c}{d} = \frac{ad+bc}{bd}$）。直接一项一项通分肯定复杂度不对，但是我们可以分治进行通分，复杂度就对了。算出来整个的结果之后算出分子*分母的逆元即可。

拉格朗日反演

- 复合逆：若多项式f和g满足$g\big(f(x)\big)=x$，那么一定也满足$f\big(g(x)\big)=x$，我们称f和g互为复合逆。
- 拉格朗日反演：若f和g互为复合逆，那么$[x^n]f(x)=\frac{1}{n}[x^{-1}]\frac{1}{g(x)^n}$。
- 扩展版：若f和g互为复合逆，那么$[x^n]h(f(x))=\frac{1}{n}[x^{-1}]\frac{h'(x)}{g(x)^n}$。
- 证明过程较为复杂，就不证了。

简单应用

- 例8，大朋友和二叉树。
- 给定s和集合D，求叶子个数为s且满足每个非叶节点的儿子个数在集合D里的多叉树个数，mod 950009857（也是一个ntt模数）
- $s, |D| \leq 10^5$。

* Source: bzoj3684

简单应用

- 设答案的生成函数为F(x)，那么我们有 $F(x)=\sum_{i\in D}F^i(x)+x$。
- 那么设D的生成函数为C(x)，我们就有$F(x)=C\big(F(x)\big)+x$，即$C\big(F(x)\big)-F(x)=x$。
- 我们令$G(x)=x-C(x)$，我们就有$G\big(F(x)\big)=x$。
- $[x^s]F(x)=\frac{1}{n}[x^{-1}]\frac{1}{G(x)^s}$，直接计算即可。

简单应用

- 例9，loj6363。
- 给定集合S，求n个点的所有极大点双连通分量点数都在S中的图数量 mod 998244353。
- 点双连通分量：删去任意一个点后剩下的点依然保持连通的连通子图。
- 极大点双连通分量：极大的点双连通分量。
- 两个简单无向图不同，当且仅当存在某条边(u,v)出现在了恰一个无向图。
- $n \le 10^5$，$\sum_{x \in S} x \le 10^5$。

* Source: loj6363

简单应用

- 首先容易求出H(x)表示带标号有根简单无向连通图的egf。无根的话只要容斥一下1所在联通块大小就可以了，有根就x^i项系数乘个i就行。
- 考虑一个有根连通图的构成，对于选中的根，我们考虑删去根之后每个连通块的贡献。每个连通块中都有一些点与根在同一个点双内，我们删去点双中所有边后，会剩下很多个以这些点为根的有根连通图。
- 我们设b[k]为k+1个点的带标号点双个数，那么我们枚举一个联通块中有k个点与根在同一个点双内，那么每个联通块egf就是：$\sum_{k\geq 1}\frac{b_k}{k!}H^k(x)$。
- 记B为b的egf，那么把每个联通块带编号并在一起就有$H(x)=xe^{B(H(x))}$。

简单应用

- $H(x)=xe^{B(H(x))}$即$\frac{H(x)}{e^{B(H(x))}}=x$，所以设$H^{-1}(x)=\frac{x}{e^{B(x)}}$。
- 我们想求$B(x)$，那么设$G(x)=\ln(\frac{H(x)}{x})$，则$G\left(H^{-1}(x)\right)=B(x)$。
- 带入拉格朗日反演式子，$[x^n]B(x)=[x^n]G\left(H^{-1}(x)\right)=\frac{1}{n}[x^{-1}]\frac{G'(x)}{H^n(x)}$。
- 那么我们就可以$O(n\log(n))$求出b_n，从而求出$B'(x)=\sum_{k\in S}\frac{b_k}{k!}x^k$。
- 那么答案的egf F同样满足$F(x)=xe^{B'(F(x))}$，$\frac{F(x)}{e^{B'(F(x))}}=x$。
- 同样$F^{-1}(x)=\frac{x}{e^{B'(x)}}$，所以$[x^n]F(x)=\frac{1}{n}[x^{-1}]\frac{1}{F^{-1}(x)^n}=\frac{1}{n}[x^{-1}]\left(\frac{e^{B'(x)}}{x}\right)^n$。

讲完了

谢谢大家