生成函数，多项式算法与图的计数

汪乐平

清华大学

2019年1月28日

多项式

由数或字母的积组成的代数式叫单项式。

多项式

由数或字母的积组成的代数式叫单项式。
有限个单项式的和叫多项式。

多项式

由数或字母的积组成的代数式叫单项式。
有限个单项式的和叫多项式。
与本课件有关的内容只有一元多项式(只包含数字和字母x)，下文中提到多项式的地方均指一元多项式。

多项式

由数或字母的积组成的代数式叫单项式。

有限个单项式的和叫多项式。

与本课件有关的内容只有一元多项式(只包含数字和字母x)，下文中提到多项式的地方均指一元多项式。

为了方便，所有多项式均表示为$\sum_{i=0}^{n} a_i x^i$的形式。定义n为这个多项式的次数。

多项式

由数或字母的积组成的代数式叫单项式。

有限个单项式的和叫多项式。

与本课件有关的内容只有一元多项式(只包含数字和字母x)，下文中提到多项式的地方均指一元多项式。

为了方便，所有多项式均表示为$\sum_{i=0}^{n} a_i x^i$的形式。定义n为这个多项式的次数。

幂级数

本页忽略级数的敛散性。数列指无穷数列。

幂级数

本页忽略级数的敛散性。数列指无穷数列。
设数列$\{a_n\}$。定义级数$S=\sum_{n\geq 0} a_n$，S叫做级数的和。

幂级数

本页忽略级数的敛散性。数列指无穷数列。

设数列$\{a_n\}$。定义级数$S = \sum_{n\geq 0} a_n$，S叫做级数的和。

设函数列$\{F_n(x)\}$。定义函数项级数$S(x) = \sum_{n\geq 0} F_n(x)$，$S(x)$叫做函数项级数的和函数。

幂级数

本页忽略级数的敛散性。数列指无穷数列。

设数列$\{a_n\}$。定义级数$S = \sum_{n\geq 0} a_n$，S叫做级数的和。

设函数列$\{F_n(x)\}$。定义函数项级数$S(x) = \sum_{n\geq 0} F_n(x)$，$S(x)$叫做函数项级数的和函数。

设$F_n(x) = a_n x^n$。定义幂级数$S(x) = \sum_{n\geq 0} a_n x^n$，$S(x)$叫做幂级数的和函数。

幂级数

本页忽略级数的敛散性。数列指无穷数列。

设数列$\{a_n\}$。定义级数$S = \sum_{n\geq 0} a_n$，S叫做级数的和。

设函数列$\{F_n(x)\}$。定义函数项级数$S(x) = \sum_{n\geq 0} F_n(x)$，$S(x)$叫做函数项级数的和函数。

设$F_n(x) = a_n x^n$。定义幂级数$S(x) = \sum_{n\geq 0} a_n x^n$，$S(x)$叫做幂级数的和函数。

通俗的说，幂级数就是无穷项的多项式。

幂级数

本页忽略级数的敛散性。数列指无穷数列。

设数列$\{a_n\}$。定义级数$S=\sum_{n\geq 0}a_n$，S叫做级数的和。

设函数列$\{F_n(x)\}$。定义函数项级数$S(x)=\sum_{n\geq 0}F_n(x)$，$S(x)$叫做函数项级数的和函数。

设$F_n(x)=a_nx^n$。定义幂级数$S(x)=\sum_{n\geq 0}a_nx^n$，$S(x)$叫做幂级数的和函数。

通俗的说，幂级数就是无穷项的多项式。

定义幂级数$S(x) \mod x^n=\sum_{i=0}^{n-1}a_ix^i$，即将幂级数截取前$n$项得到一个$n-1$次多项式。

生成函数

终于说到生成函数了。

生成函数

终于说到生成函数了。

生成函数是用于对应一个无穷序列的幂级数。在图的计数中能用到的一般是三种。

生成函数

终于说到生成函数了。

生成函数是用于对应一个无穷序列的幂级数。在图的计数中能用到的一般是三种。

$\{a_n\}$的普通生成函数$O(x) = \sum_{n\geq 0} a_n x^n$。一般在无编号的计数题中使用。

生成函数

终于说到生成函数了。

生成函数是用于对应一个无穷序列的幂级数。在图的计数中能用到的一般是三种。

$\{a_n\}$的普通生成函数$O(x) = \sum_{n\geq 0} a_n x^n$。一般在无编号的计数题中使用。

$\{a_n\}$的指数生成函数$E(x) = \sum_{n\geq 0} \frac{a_n}{n!} x^n$。一般在有编号的计数题中使用。

生成函数

终于说到生成函数了。

生成函数是用于对应一个无穷序列的幂级数。在图的计数中能用到的一般是三种。

$\{a_n\}$的普通生成函数$O(x) = \sum_{n\geq 0} a_n x^n$。一般在无编号的计数题中使用。

$\{a_n\}$的指数生成函数$E(x) = \sum_{n\geq 0} \frac{a_n}{n!} x^n$。一般在有编号的计数题中使用。

$\{a_n\}$的组合生成函数$S(x) = \sum_{n\geq 0} \frac{a_n}{n! 2^{\binom{n}{2}}} x^n$。一般在有编号有向图的计数题中使用。

生成函数

终于说到生成函数了。

生成函数是用于对应一个无穷序列的幂级数。在图的计数中能用到的一般是三种。

$\{a_n\}$的普通生成函数$O(x)=\sum_{n\geq 0} a_n x^n$。一般在无编号的计数题中使用。

$\{a_n\}$的指数生成函数$E(x)=\sum_{n\geq 0} \frac{a_n}{n!} x^n$。一般在有编号的计数题中使用。

$\{a_n\}$的组合生成函数$S(x)=\sum_{n\geq 0} \frac{a_n}{n!2^{\binom{n}{2}}} x^n$。一般在有编号有向图的计数题中使用。

由于无编号的图较难计数，而有向图计数的研究有限，因此目前最常用的生成函数是指数生成函数。

生成函数四则运算

数列可以对应到生成函数，那么生成函数运算后便可以得到一个新的数列的生成函数。

生成函数四则运算

数列可以对应到生成函数，那么生成函数运算后便可以得到一个新的数列的生成函数。

设数列$\{a_n\},\{b_n\}$，对应的生成函数是$A(x), B(x)$。

生成函数四则运算

数列可以对应到生成函数，那么生成函数运算后便可以得到一个新的数列的生成函数。

设数列$\{a_n\},\{b_n\}$，对应的生成函数是$A(x),B(x)$。

$A(x)\pm B(x)$是数列$\{a_n\pm b_n\}$对应的生成函数。

生成函数四则运算

数列可以对应到生成函数，那么生成函数运算后便可以得到一个新的数列的生成函数。

设数列$\{a_n\}, \{b_n\}$，对应的生成函数是$A(x), B(x)$。

$A(x) \pm B(x)$是数列$\{a_n \pm b_n\}$对应的生成函数。

$A(x)B(x)$对应的数列需要看$A(x), B(x)$的种类。

生成函数四则运算

数列可以对应到生成函数，那么生成函数运算后便可以得到一个新的数列的生成函数。

设数列$\{a_n\},\{b_n\}$，对应的生成函数是$A(x),B(x)$。

$A(x)\pm B(x)$是数列$\{a_n \pm b_n\}$对应的生成函数。

$A(x)B(x)$对应的数列需要看$A(x),B(x)$的种类。

普通生成函数：$A(x)B(x)$是数列$\{\sum_{i=0}^{n} a_i b_{n-i}\}$对应的普通生成函数。

生成函数四则运算

数列可以对应到生成函数，那么生成函数运算后便可以得到一个新的数列的生成函数。

设数列$\{a_n\},\{b_n\}$，对应的生成函数是$A(x),B(x)$。

$A(x)\pm B(x)$是数列$\{a_n \pm b_n\}$对应的生成函数。

$A(x)B(x)$对应的数列需要看$A(x),B(x)$的种类。

普通生成函数：$A(x)B(x)$是数列$\{\sum_{i=0}^{n} a_i b_{n-i}\}$对应的普通生成函数。

指数生成函数：$A(x)B(x)$是数列$\{\sum_{i=0}^{n} \binom{n}{i} a_i b_{n-i}\}$对应的指数生成函数。

生成函数四则运算

数列可以对应到生成函数，那么生成函数运算后便可以得到一个新的数列的生成函数。

设数列$\{a_n\}, \{b_n\}$，对应的生成函数是$A(x), B(x)$。

$A(x) \pm B(x)$是数列$\{a_n \pm b_n\}$对应的生成函数。

$A(x)B(x)$对应的数列需要看$A(x), B(x)$的种类。

普通生成函数：$A(x)B(x)$是数列$\{\sum_{i=0}^{n} a_i b_{n-i}\}$对应的普通生成函数。

指数生成函数：$A(x)B(x)$是数列$\{\sum_{i=0}^{n} \binom{n}{i} a_i b_{n-i}\}$对应的指数生成函数。

组合生成函数：$A(x)B(x)$是数列$\{\sum_{i=0}^{n} \binom{n}{i} 2^{i(n-i)} a_i b_{n-i}\}$对应的指数生成函数。

生成函数四则运算

数列可以对应到生成函数，那么生成函数运算后便可以得到一个新的数列的生成函数。

设数列$\{a_n\},\{b_n\}$，对应的生成函数是$A(x),B(x)$。

$A(x)\pm B(x)$是数列$\{a_n\pm b_n\}$对应的生成函数。

$A(x)B(x)$对应的数列需要看$A(x),B(x)$的种类。

普通生成函数：$A(x)B(x)$是数列$\{\sum_{i=0}^{n} a_i b_{n-i}\}$对应的普通生成函数。

指数生成函数：$A(x)B(x)$是数列$\{\sum_{i=0}^{n}\binom{n}{i} a_i b_{n-i}\}$对应的指数生成函数。

组合生成函数：$A(x)B(x)$是数列$\{\sum_{i=0}^{n}\binom{n}{i} 2^{i(n-i)} a_i b_{n-i}\}$对应的指数生成函数。

生成函数除法对应的数列较为复杂。实际应用中一般是先将递推式写成生成函数乘法的形式，再移项得到生成函数除法。

求导与积分

导数与积分较为复杂，在此仅给出幂级数求导与积分的结果。设$A(x) = \sum_{n\geq 0} a_n x^n$。

求导与积分

导数与积分较为复杂，在此仅给出幂级数求导与积分的结果。
设$A(x) = \sum_{n\geq 0} a_n x^n$。
$A(x)$的导函数$A'(x) = \sum_{n\geq 0} (n+1)a_{n+1}x^n$。

求导与积分

导数与积分较为复杂，在此仅给出幂级数求导与积分的结果。

设$A(x) = \sum_{n\geq 0} a_n x^n$。

$A(x)$的导函数$A'(x) = \sum_{n\geq 0} (n+1)a_{n+1}x^n$。

$A(x)$的积分函数$\int A(x)\mathrm{d}x = \sum_{n\geq 0} \frac{a_n}{n+1}x^{n+1}$(默认常数项为0)。

求导与积分

导数与积分较为复杂，在此仅给出幂级数求导与积分的结果。
设$A(x)=\sum_{n\geq 0}a_nx^n$。

$A(x)$的导函数$A'(x)=\sum_{n\geq 0}(n+1)a_{n+1}x^n$。

$A(x)$的积分函数$\int A(x)\mathrm{d}x=\sum_{n\geq 0}\frac{a_n}{n+1}x^{n+1}$(默认常数项为0)。

设$E(x)$是数列$\{a_n\}$对应的指数生成函数，那么$E'(x)$是数列$\{a_{n+1}\}$对应的指数生成函数，$\int E(x)\mathrm{d}x$是数列$\{a_{n-1}\}$ 对应的指数生成函数。

ln与exp

幂级数ln：$\ln A(x) = \int \frac{A'(x)}{A(x)} \mathrm{d}x$。

ln 与 exp

幂级数ln：$\ln A(x) = \int \frac{A'(x)}{A(x)} \mathrm{d}x$。
即在$A(x)$已知时，方程$B'(x)A(x) = A'(x)$的解是$B(x) = \ln A(x)$。

ln与exp

幂级数ln：$\ln A(x) = \int \frac{A'(x)}{A(x)} \mathrm{d}x$。
即在$A(x)$已知时，方程$B'(x)A(x) = A'(x)$的解是$B(x) = \ln A(x)$。
在$B(x)$已知时，方程$B'(x)A(x) = A'(x)$的解是$A(x) = \exp B(x)$。

ln与exp

幂级数ln：$\ln A(x) = \int \frac{A'(x)}{A(x)} \mathrm{d}x$。
即在$A(x)$已知时，方程$B'(x)A(x) = A'(x)$的解是$B(x) = \ln A(x)$。
在$B(x)$已知时，方程$B'(x)A(x) = A'(x)$的解是$A(x) = \exp B(x)$。
ln一般在已知不同集合个数求不同元素个数时使用。

ln与exp

幂级数ln：$\ln A(x) = \int \frac{A'(x)}{A(x)} \mathrm{d}x$。
即在$A(x)$已知时，方程$B'(x)A(x) = A'(x)$的解是$B(x) = \ln A(x)$。
在$B(x)$已知时，方程$B'(x)A(x) = A'(x)$的解是$A(x) = \exp B(x)$。
ln一般在已知不同集合个数求不同元素个数时使用。
exp一般在已知不同元素个数求不同集合个数时使用。

ln与exp

幂级数ln：$\ln A(x) = \int \frac{A'(x)}{A(x)} \mathrm{d}x$。
即在$A(x)$已知时，方程$B'(x)A(x) = A'(x)$的解是$B(x) = \ln A(x)$。
在$B(x)$已知时，方程$B'(x)A(x) = A'(x)$的解是$A(x) = \exp B(x)$。
ln一般在已知不同集合个数求不同元素个数时使用。
exp一般在已知不同元素个数求不同集合个数时使用。
可以使用ln和exp计算$F(x)^k$。

幂级数复合与复合逆

设$A(x)=\sum_{n\geq 1}a_nx^n$，$B(x)=\sum_{n\geq 1}b_nx^n$。

幂级数复合与复合逆

设$A(x)=\sum_{n\geq 1}a_nx^n$，$B(x)=\sum_{n\geq 1}b_nx^n$。
则$A(B(x))=\sum_{n\geq 1}a_nB(x)^n$，是函数项级数，可以展开成幂级数。

幂级数复合与复合逆

设$A(x) = \sum_{n\geq 1} a_n x^n$，$B(x) = \sum_{n\geq 1} b_n x^n$。
则$A(B(x)) = \sum_{n\geq 1} a_n B(x)^n$，是函数项级数，可以展开成幂级数。
如果$A(B(x)) = B(A(x)) = x$，那么$A(x)$和$B(x)$互为复合逆。

幂级数复合与复合逆

设$A(x) = \sum_{n\geq 1} a_n x^n$，$B(x) = \sum_{n\geq 1} b_n x^n$。
则$A(B(x)) = \sum_{n\geq 1} a_n B(x)^n$，是函数项级数，可以展开成幂级数。
如果$A(B(x)) = B(A(x)) = x$，那么$A(x)$和$B(x)$互为复合逆。
在结构具有层次性的计数题中使用较多。

多项式算法

使用生成函数时，很多时候并不需要找出普遍规律，只需要计算数列中的有限项，即求出一个多项式即可。

多项式算法

使用生成函数时，很多时候并不需要找出普遍规律，只需要计算数列中的有限项，即求出一个多项式即可。

因此考虑如何对多项式使用各种运算。以下设多项式次数为n。

多项式算法

使用生成函数时，很多时候并不需要找出普遍规律，只需要计算数列中的有限项，即求出一个多项式即可。

因此考虑如何对多项式使用各种运算。以下设多项式次数为n。

多项式加减法直接对每一项进行加减即可，时间复杂度$O(n)$。

多项式乘法

点值表示法：将n次多项式$A(x)$用$n+1$个不同的$A(x_i)=y_i$表示。

多项式乘法

点值表示法：将n次多项式$A(x)$用$n+1$个不同的$A(x_i)=y_i$表示。

利用额外的点值，点值表示法可以在$O(n)$的时间复杂度内计算两个多项式的乘法。

多项式乘法

点值表示法：将n次多项式$A(x)$用$n+1$个不同的$A(x_i)=y_i$表示。

利用额外的点值，点值表示法可以在$O(n)$的时间复杂度内计算两个多项式的乘法。

问题在于如何快速在系数表示法和点值表示法之间互相转化。

多项式乘法

点值表示法：将n次多项式$A(x)$用$n+1$个不同的$A(x_i)=y_i$表示。

利用额外的点值，点值表示法可以在$O(n)$的时间复杂度内计算两个多项式的乘法。

问题在于如何快速在系数表示法和点值表示法之间互相转化。

为了方便，假设多项式的系数均属于 $\bmod 998244353$的剩余系($998244353=7*17*2^{23}+1$，是质数)。

快速数论变换

质数P的原根g：$\forall 1 \le x < P, \exists 0 \le y < P-1, g^y \equiv x \pmod{P}$。

快速数论变换

质数P的原根g：$\forall 1 \le x < P, \exists 0 \le y < P-1, g^y \equiv x \pmod P$。
设$\omega_m = g^{\frac{P-1}{2^m}}$，$A(x) = A_0(x^2) + xA_1(x^2)$。

快速数论变换

质数P的原根g：$\forall 1 \le x < P, \exists 0 \le y < P-1, g^y \equiv x (\mod P)$。
设$\omega_m = g^{\frac{P-1}{2^m}}$，$A(x) = A_0(x^2) + xA_1(x^2)$。
$A(\omega_m^k) = A_0(\omega_{m-1}^k) + \omega_m^k A_1(\omega_{m-1}^k)$

快速数论变换

质数P的原根g：$\forall 1 \le x < P, \exists 0 \le y < P-1, g^y \equiv x \pmod P$。
设$\omega_m = g^{\frac{P-1}{2^m}}$，$A(x) = A_0(x^2) + xA_1(x^2)$。
$A(\omega_m^k) = A_0(\omega_{m-1}^k) + \omega_m^k A_1(\omega_{m-1}^k)$
$A(\omega_m^{k+2^{m-1}}) = A_0(\omega_{m-1}^k) - \omega_m^k A_1(\omega_{m-1}^k)$

快速数论变换

质数P的原根g：$\forall 1 \le x < P, \exists 0 \le y < P-1, g^y \equiv x \pmod P$。
设$\omega_m = g^{\frac{P-1}{2^m}}$，$A(x) = A_0(x^2) + xA_1(x^2)$。
$A(\omega_m^k) = A_0(\omega_{m-1}^k) + \omega_m^k A_1(\omega_{m-1}^k)$
$A(\omega_m^{k+2^{m-1}}) = A_0(\omega_{m-1}^k) - \omega_m^k A_1(\omega_{m-1}^k)$
因此可以使用分治算法，初始取$2^{m-1} \le n < 2^m$即可。时间复杂度$T(n) = 2T(n/2) + O(n)$，解得$T(n) = O(n \log n)$。

快速数论变换

质数P的原根g：$\forall 1 \le x < P, \exists 0 \le y < P-1, g^y \equiv x \pmod P$。
设$\omega_m = g^{\frac{P-1}{2^m}}$，$A(x) = A_0(x^2) + xA_1(x^2)$。
$A(\omega_m^k) = A_0(\omega_{m-1}^k) + \omega_m^k A_1(\omega_{m-1}^k)$
$A(\omega_m^{k+2^{m-1}}) = A_0(\omega_{m-1}^k) - \omega_m^k A_1(\omega_{m-1}^k)$
因此可以使用分治算法，初始取$2^{m-1} \le n < 2^m$即可。时间复杂度$T(n) = 2T(n/2) + O(n)$，解得$T(n) = O(n \log n)$。
点值表示法转回系数表示法的方法是类似的，在此略去。

多项式逆元

已知n次多项式$P(x)$，$P(0)=1$，求幂级数$F(x)$满足$P(x)F(x)=1$。

多项式逆元

已知n次多项式$P(x)$，$P(0) = 1$，求幂级数$F(x)$满足$P(x)F(x) = 1$。
设$F_t(x) = F(x) \mod x^{2^t}$。

多项式逆元

已知n次多项式$P(x)$，$P(0)=1$，求幂级数$F(x)$满足$P(x)F(x)=1$。
设$F_t(x)=F(x) \mod x^{2^t}$。
$F_0(x)=F(x) \mod x = F(0)=\frac{1}{P(0)}=1$

多项式逆元

已知n次多项式$P(x)$，$P(0)=1$，求幂级数$F(x)$满足$P(x)F(x)=1$。
设$F_t(x)=F(x) \mod x^{2^t}$。
$F_0(x)=F(x) \mod x = F(0)=\frac{1}{P(0)}=1$
若方程为$G(F(x))=0$，则$F_{t+1}(x)\equiv F_t(x)-\frac{G(F_t(x))}{G'(F_t(x))}(\mod x^{2^{t+1}})$。

多项式逆元

已知n次多项式$P(x)$，$P(0)=1$，求幂级数$F(x)$满足$P(x)F(x)=1$。
设$F_t(x)=F(x) \mod x^{2^t}$。
$F_0(x)=F(x) \mod x = F(0)=\frac{1}{P(0)}=1$
若方程为$G(F(x))=0$，则$F_{t+1}(x)\equiv F_t(x)-\frac{G(F_t(x))}{G'(F_t(x))} (\mod x^{2^{t+1}})$。
解得$F_{t+1}(x)\equiv F_t(x)(2-P(x)F_t(x)) (\mod x^{2^{t+1}})$。

多项式逆元

已知n次多项式$P(x)$，$P(0)=1$，求幂级数$F(x)$满足$P(x)F(x)=1$。
设$F_t(x)=F(x) \mod x^{2^t}$。
$F_0(x)=F(x) \mod x = F(0)=\frac{1}{P(0)}=1$
若方程为$G(F(x))=0$，则$F_{t+1}(x)\equiv F_t(x)-\frac{G(F_t(x))}{G'(F_t(x))} (\mod x^{2^{t+1}})$。
解得$F_{t+1}(x)\equiv F_t(x)(2-P(x)F_t(x)) (\mod x^{2^{t+1}})$。
时间复杂度$T(n)=T(n/2)+O(n\log n)$，解得$T(n)=O(n\log n)$。

多项式ln与exp

$\ln F(x) = \int \frac{F'(x)}{F(x)} \mathrm{d}x$，直接计算即可，时间复杂度$O(n \log n)$。

多项式ln与exp

$\ln F(x) = \int \frac{F'(x)}{F(x)} \mathrm{d}x$，直接计算即可，时间复杂度$O(n \log n)$。
exp的方程是$F(x) = \exp P(x)$，ln并移项得$\ln F(x) - P(x) = 0$。

多项式ln与exp

$\ln F(x) = \int \frac{F'(x)}{F(x)} \mathrm{d}x$，直接计算即可，时间复杂度$O(n \log n)$。
exp的方程是$F(x) = \exp P(x)$，ln并移项得$\ln F(x) - P(x) = 0$。
牛顿迭代解得$F_{t+1}(x) \equiv F_t(x)(1 - \ln F_t(x) + P(x)) \pmod{x^{2^{t+1}}}$。

多项式ln与exp

$\ln F(x) = \int \frac{F'(x)}{F(x)}\mathrm{d}x$，直接计算即可，时间复杂度$O(n\log n)$。
exp的方程是$F(x) = \exp P(x)$，ln并移项得$\ln F(x) - P(x) = 0$。
牛顿迭代解得$F_{t+1}(x) \equiv F_t(x)(1 - \ln F_t(x) + P(x))(\mod x^{2^{t+1}})$。
时间复杂度$T(n) = T(n/2) + O(n\log n)$，解得$T(n) = O(n\log n)$。
利用ln和exp可以在$O(n\log n)$的时间复杂度内计算$P(x)^k$。

多项式复合逆

很遗憾，复合逆没有时间复杂度$O(n\log n)$的算法。

多项式复合逆

很遗憾，复合逆没有时间复杂度$O(n\log n)$的算法。

但是可以使用拉格朗日反演在$O(n\log n)$时间复杂度内算出一项。
设$[x^n]F(x)$表示$F(x)$的x^n项的系数。

多项式复合逆

很遗憾，复合逆没有时间复杂度$O(n\log n)$的算法。

但是可以使用拉格朗日反演在$O(n\log n)$时间复杂度内算出一项。设$[x^n]F(x)$表示$F(x)$的x^n项的系数。

设$F(G(x)) = G(F(x)) = x$，则$[x^n]G(x) = \frac{1}{n}[x^{n-1}](\frac{x}{F(x)})^n$。使用之前的算法即可。

多项式复合逆

很遗憾，复合逆没有时间复杂度$O(n\log n)$的算法。

但是可以使用拉格朗日反演在$O(n\log n)$时间复杂度内算出一项。设$[x^n]F(x)$表示$F(x)$的x^n项的系数。

设$F(G(x))=G(F(x))=x$，则$[x^n]G(x)=\frac{1}{n}[x^{n-1}](\frac{x}{F(x)})^n$。使用之前的算法即可。

还有一个扩展形式$[x^n]H(G(x))=\frac{1}{n}[x^{n-1}]H'(x)(\frac{x}{F(x)})^n$。

多项式带余除法

给出n次多项式$A(x)$和m次多项式$B(x)(n \geq m)$，求$C(x), D(x)$使得$A(x) = B(x)C(x) + D(x)$且$D(x)$的次数$< m$。

多项式带余除法

给出n次多项式$A(x)$和m次多项式$B(x)(n \geq m)$，求$C(x), D(x)$使得$A(x) = B(x)C(x) + D(x)$且$D(x)$的次数$< m$。

$x^n A(\frac{1}{x}) = x^m B(\frac{1}{x}) x^{n-m} C(\frac{1}{x}) + x^n D(\frac{1}{x})$

多项式带余除法

给出n次多项式$A(x)$和m次多项式$B(x)(n \geq m)$，求$C(x), D(x)$使得$A(x) = B(x)C(x) + D(x)$且$D(x)$的次数$< m$。

$$x^n A(\frac{1}{x}) = x^m B(\frac{1}{x}) x^{n-m} C(\frac{1}{x}) + x^n D(\frac{1}{x})$$
$$x^n A(\frac{1}{x}) \equiv x^m B(\frac{1}{x}) x^{n-m} C(\frac{1}{x}) \pmod{x^{n-m+1}}$$

多项式带余除法

给出n次多项式$A(x)$和m次多项式$B(x)(n \geq m)$，求$C(x), D(x)$使得$A(x) = B(x)C(x) + D(x)$且$D(x)$的次数$< m$。

$$x^n A(\frac{1}{x}) = x^m B(\frac{1}{x}) x^{n-m} C(\frac{1}{x}) + x^n D(\frac{1}{x})$$

$$x^n A(\frac{1}{x}) \equiv x^m B(\frac{1}{x}) x^{n-m} C(\frac{1}{x}) (\mod x^{n-m+1})$$

因此使用一次多项式求逆和多项式乘法即可得到$C(x)$。

多项式带余除法

给出n次多项式$A(x)$和m次多项式$B(x)(n \geq m)$，求$C(x), D(x)$使得$A(x) = B(x)C(x) + D(x)$且$D(x)$的次数$< m$。

$x^n A(\frac{1}{x}) = x^m B(\frac{1}{x}) x^{n-m} C(\frac{1}{x}) + x^n D(\frac{1}{x})$

$x^n A(\frac{1}{x}) \equiv x^m B(\frac{1}{x}) x^{n-m} C(\frac{1}{x}) (\mod x^{n-m+1})$

因此使用一次多项式求逆和多项式乘法即可得到$C(x)$。

$D(x) = A(x) - B(x)C(x)$

多项式带余除法

给出n次多项式$A(x)$和m次多项式$B(x)(n \geq m)$，求$C(x), D(x)$使得$A(x) = B(x)C(x) + D(x)$且$D(x)$的次数$< m$。

$x^n A(\frac{1}{x}) = x^m B(\frac{1}{x}) x^{n-m} C(\frac{1}{x}) + x^n D(\frac{1}{x})$

$x^n A(\frac{1}{x}) \equiv x^m B(\frac{1}{x}) x^{n-m} C(\frac{1}{x}) (\mod x^{n-m+1})$

因此使用一次多项式求逆和多项式乘法即可得到$C(x)$。

$D(x) = A(x) - B(x)C(x)$

时间复杂度$O(n \log n)$。

基本的指数生成函数

以下均为有标号图。

基本的指数生成函数

以下均为有标号图。

树：$A(x) = \sum_{n\geq 0} \frac{n^{n-2}}{n!} x^n$

基本的指数生成函数

以下均为有标号图。
树：$A(x) = \sum_{n\geq 0} \frac{n^{n-2}}{n!}x^n$
森林：$B(x) = \exp A(x)$

基本的指数生成函数

以下均为有标号图。
树：$A(x)=\sum_{n\geq 0}\frac{n^{n-2}}{n!}x^n$
森林：$B(x)=\exp A(x)$
无向图：$C(x)=\sum_{n\geq 0}\frac{2^{\binom{n}{2}}}{n!}x^n$

基本的指数生成函数

以下均为有标号图。
树：$A(x) = \sum_{n\geq 0} \frac{n^{n-2}}{n!} x^n$
森林：$B(x) = \exp A(x)$
无向图：$C(x) = \sum_{n\geq 0} \frac{2^{\binom{n}{2}}}{n!} x^n$
无向连通图：$D(x) = \ln C(x)$

基本的指数生成函数

以下均为有标号图。
树：$A(x) = \sum_{n\geq 0} \frac{n^{n-2}}{n!} x^n$
森林：$B(x) = \exp A(x)$
无向图：$C(x) = \sum_{n\geq 0} \frac{2^{\binom{n}{2}}}{n!} x^n$
无向连通图：$D(x) = \ln C(x)$

小朋友与二叉树

给一个含有n个正整数的集合C和一个正整数m。

小朋友与二叉树

给一个含有n个正整数的集合C和一个正整数m。

一棵带权二叉树是合法的当且仅当树中每一个结点的权值都在集合C内。

小朋友与二叉树

给一个含有n个正整数的集合C和一个正整数m。

一棵带权二叉树是合法的当且仅当树中每一个结点的权值都在集合C内。

定义二叉树的权值是二叉树中所有结点权值的和。

小朋友与二叉树

给一个含有n个正整数的集合C和一个正整数m。

一棵带权二叉树是合法的当且仅当树中每一个结点的权值都在集合C内。

定义二叉树的权值是二叉树中所有结点权值的和。

对于$\forall 1 \leq s \leq m$，询问有多少棵不同的权值为s的二叉树。

小朋友与二叉树

给一个含有n个正整数的集合C和一个正整数m。

一棵带权二叉树是合法的当且仅当树中每一个结点的权值都在集合C内。

定义二叉树的权值是二叉树中所有结点权值的和。

对于$\forall 1 \le s \le m$，询问有多少棵不同的权值为s的二叉树。

两棵二叉树是相同的当且仅当结构相同且相同位置上的结点权值相同。

小朋友与二叉树

给一个含有n个正整数的集合C和一个正整数m。

一棵带权二叉树是合法的当且仅当树中每一个结点的权值都在集合C内。

定义二叉树的权值是二叉树中所有结点权值的和。

对于$\forall 1 \leq s \leq m$，询问有多少棵不同的权值为s的二叉树。

两棵二叉树是相同的当且仅当结构相同且相同位置上的结点权值相同。

所有数字均不超过10^5。

小朋友与二叉树——解答

设$G(x) = \sum_{n\geq 0} [n \in C] x^n$，合法二叉树对应的普通生成函数为$F(x)$。

小朋友与二叉树——解答

设$G(x)=\sum_{n\geq 0}[n\in C]x^n$，合法二叉树对应的普通生成函数为$F(x)$。

依据题目列出方程$F(x)=F(x)^2G(x)+1$，移项得$G(x)F(x)^2-F(x)+1=0$

小朋友与二叉树——解答

设$G(x)=\sum_{n\geq 0}[n\in C]x^n$，合法二叉树对应的普通生成函数为$F(x)$。

依据题目列出方程$F(x)=F(x)^2G(x)+1$，移项得$G(x)F(x)^2-F(x)+1=0$

解方程得$F(x)=\frac{1\pm\sqrt{1-4G(x)}}{2G(x)}=\frac{2}{1\mp\sqrt{1-4G(x)}}$

小朋友与二叉树——解答

设$G(x)=\sum_{n\geq 0}[n\in C]x^n$，合法二叉树对应的普通生成函数为$F(x)$。

依据题目列出方程$F(x)=F(x)^2G(x)+1$，移项得$G(x)F(x)^2-F(x)+1=0$

解方程得$F(x)=\frac{1\pm\sqrt{1-4G(x)}}{2G(x)}=\frac{2}{1\mp\sqrt{1-4G(x)}}$

分母常数项不能为0，因此$F(x)=\frac{2}{1+\sqrt{1-4G(x)}}$

小朋友与二叉树——解答

设$G(x)=\sum_{n\geq 0}[n\in C]x^n$，合法二叉树对应的普通生成函数为$F(x)$。

依据题目列出方程$F(x)=F(x)^2G(x)+1$，移项得$G(x)F(x)^2-F(x)+1=0$

解方程得$F(x)=\frac{1\pm\sqrt{1-4G(x)}}{2G(x)}=\frac{2}{1\mp\sqrt{1-4G(x)}}$

分母常数项不能为0，因此$F(x)=\frac{2}{1+\sqrt{1-4G(x)}}$

调用多项式幂和多项式求逆即可，时间复杂度$O(n\log n)$。

小朋友与二叉树——解答

设$G(x) = \sum_{n\geq 0}[n \in C]x^n$，合法二叉树对应的普通生成函数为$F(x)$。

依据题目列出方程$F(x) = F(x)^2G(x) + 1$，移项得$G(x)F(x)^2 - F(x) + 1 = 0$

解方程得$F(x) = \frac{1\pm\sqrt{1-4G(x)}}{2G(x)} = \frac{2}{1\mp\sqrt{1-4G(x)}}$

分母常数项不能为0，因此$F(x) = \frac{2}{1+\sqrt{1-4G(x)}}$

调用多项式幂和多项式求逆即可，时间复杂度$O(n\log n)$。

当时还发明了一个多项式$sqrt$。

大朋友与多叉树

给一个含有m个正整数的集合D和一个正整数$s(m < s, 1 \notin D)$。

大朋友与多叉树

给一个含有m个正整数的集合D和一个正整数$s(m < s, 1 \notin D)$。
一棵树是合法的当且仅当每个非叶结点的孩子数量都在集合D内。

大朋友与多叉树

给一个含有m个正整数的集合D和一个正整数$s(m < s, 1 \notin D)$。
一棵树是合法的当且仅当每个非叶结点的孩子数量都在集合D内。
询问有多少棵不同的含有s个叶结点的树。

大朋友与多叉树

给一个含有m个正整数的集合D和一个正整数$s(m < s, 1 \notin D)$。
一棵树是合法的当且仅当每个非叶结点的孩子数量都在集合D内。
询问有多少棵不同的含有s个叶结点的树。
两棵树是相同的当且仅当结构相同(孩子之间有顺序)。

大朋友与多叉树

给一个含有m个正整数的集合D和一个正整数$s(m < s, 1 \notin D)$。
一棵树是合法的当且仅当每个非叶结点的孩子数量都在集合D内。
询问有多少棵不同的含有s个叶结点的树。
两棵树是相同的当且仅当结构相同(孩子之间有顺序)。
所有数字均不超过10^5。

大朋友与多叉树——解答

设$G(x) = \sum_{n \geq 0} [n \in D] x^n$，合法树对应的普通生成函数为$F(x)$。

大朋友与多叉树——解答

设$G(x) = \sum_{n\geq 0} [n \in D]x^n$，合法树对应的普通生成函数为$F(x)$。
依据题目列出方程$F(x) = G(F(x)) + x$，移项得$F(x) - G(F(x)) = x$

大朋友与多叉树——解答

设$G(x) = \sum_{n \geq 0} [n \in D] x^n$，合法树对应的普通生成函数为$F(x)$。
依据题目列出方程$F(x) = G(F(x)) + x$，移项得$F(x) - G(F(x)) = x$
因此$F(x)$是$x - G(x)$的复合逆。

大朋友与多叉树——解答

设$G(x) = \sum_{n \geq 0} [n \in D] x^n$，合法树对应的普通生成函数为$F(x)$。
依据题目列出方程$F(x) = G(F(x)) + x$，移项得$F(x) - G(F(x)) = x$
因此$F(x)$是$x - G(x)$的复合逆。
用拉格朗日反演公式求出$[x^s]F(x)$即可，时间复杂度$O(n \log n)$。

YJC plays Minecraft

一张图由n个部分组成，第$i(1 \le i \le n)$个部分是一个a_i个点的完全图。

YJC plays Minecraft

一张图由n个部分组成，第$i(1 \le i \le n)$个部分是一个a_i个点的完全图。

第$i(1 \le i \le n)$个部分的第a_i个点和第$i \mod n + 1$个部分的第1个点连一条边。

YJC plays Minecraft

一张图由n个部分组成，第$i(1 \le i \le n)$个部分是一个a_i个点的完全图。

第$i(1 \le i \le n)$个部分的第a_i个点和第$i \mod n+1$个部分的第1个点连一条边。

求图的生成森林个数。

YJC plays Minecraft

一张图由n个部分组成，第$i(1 \le i \le n)$个部分是一个a_i个点的完全图。

第$i(1 \le i \le n)$个部分的第a_i个点和第$i \mod n+1$个部分的第1个点连一条边。

求图的生成森林个数。

所有数字均不超过10^5。

YJC plays Minecraft——解答

设f_n表示n个节点完全图生成树个数，g_n表示n个节点完全图生成森林个数，h_n表示n个节点完全图，点1和点n不在同一棵树中的生成森林个数。$F(x), G(x), H(x)$分别表示$\{f_n\}, \{g_n\}, \{h_n\}$的指数生成函数。

设f_n表示n个节点完全图生成树个数，g_n表示n个节点完全图生成森林个数，h_n表示n个节点完全图，点1和点n不在同一棵树中的生成森林个数。$F(x), G(x), H(x)$分别表示$\{f_n\}, \{g_n\}, \{h_n\}$的指数生成函数。

答案就是$2^n \prod_{i=1}^{n} g_{a_i} - \prod_{i=1}^{n}(g_{a_i} - h_{a_i})$。第一部分表示所有完全图生成森林个数乘起来再乘上大环上的边的每一种情况，第二部分表示大环上每一条边都不删，而每一个完全图中第一个点和最后一个点都在同一棵树里面的情况，这种情况是不合法的，要从答案里面减掉。

YJC plays Minecraft——解答

设f_n表示n个节点完全图生成树个数，g_n表示n个节点完全图生成森林个数，h_n表示n个节点完全图，点1和点n不在同一棵树中的生成森林个数。$F(x), G(x), H(x)$分别表示$\{f_n\}, \{g_n\}, \{h_n\}$的指数生成函数。

答案就是$2^n \prod_{i=1}^{n} g_{a_i} - \prod_{i=1}^{n}(g_{a_i} - h_{a_i})$。第一部分表示所有完全图生成森林个数乘起来再乘上大环上的边的每一种情况，第二部分表示大环上每一条边都不删，而每一个完全图中第一个点和最后一个点都在同一棵树里面的情况，这种情况是不合法的，要从答案里面减掉。

之前已经知道了$G(x) = \exp F(x)$，现在考虑$H(x)$怎么求。

设f_n表示n个节点完全图生成树个数，g_n表示n个节点完全图生成森林个数，h_n表示n个节点完全图，点1和点n不在同一棵树中的生成森林个数。$F(x), G(x), H(x)$分别表示$\{f_n\}, \{g_n\}, \{h_n\}$的指数生成函数。

答案就是$2^n\prod_{i=1}^n g_{a_i} - \prod_{i=1}^n (g_{a_i} - h_{a_i})$。第一部分表示所有完全图生成森林个数乘起来再乘上大环上的边的每一种情况，第二部分表示大环上每一条边都不删，而每一个完全图中第一个点和最后一个点都在同一棵树里面的情况，这种情况是不合法的，要从答案里面减掉。

之前已经知道了$G(x) = \exp F(x)$，现在考虑$H(x)$怎么求。

枚举1号点所在树点数可得递推式$h_n = \sum_{i=0}^{n-2} f_{i+1} g_{n-i-1} \binom{n-2}{i}$。生成函数形式为$H''(x) = F'(x)G'(x)$，即$H(x) = \int\int F'(x)G'(x)\mathrm{d}x\mathrm{d}x$。

YJC plays Minecraft——解答

设f_n表示n个节点完全图生成树个数，g_n表示n个节点完全图生成森林个数，h_n表示n个节点完全图，点1和点n不在同一棵树中的生成森林个数。$F(x), G(x), H(x)$分别表示$\{f_n\}, \{g_n\}, \{h_n\}$的指数生成函数。

答案就是$2^n \prod_{i=1}^{n} g_{a_i} - \prod_{i=1}^{n}(g_{a_i} - h_{a_i})$。第一部分表示所有完全图生成森林个数乘起来再乘上大环上的边的每一种情况，第二部分表示大环上每一条边都不删，而每一个完全图中第一个点和最后一个点都在同一棵树里面的情况，这种情况是不合法的，要从答案里面减掉。

之前已经知道了$G(x) = \exp F(x)$，现在考虑$H(x)$怎么求。

枚举1号点所在树点数可得递推式$h_n = \sum_{i=0}^{n-2} f_{i+1} g_{n-i-1} \binom{n-2}{i}$。生成函数形式为$H''(x) = F'(x)G'(x)$，即$H(x) = \int\int F'(x)G'(x)\mathrm{d}x\mathrm{d}x$。

时间复杂度$O(n \log n)$。

仙人掌

如果一个无向连通图的任意一条边最多属于一个简单环，我们就称之为仙人掌。所谓简单环即不经过重复的结点的环。

仙人掌

如果一个无向连通图的任意一条边最多属于一个简单环，我们就称之为仙人掌。所谓简单环即不经过重复的结点的环。

给一个n，对于每一个$1 \leq i \leq n$，求i个点的不同的有标号仙人掌的个数

仙人掌

如果一个无向连通图的任意一条边最多属于一个简单环，我们就称之为仙人掌。所谓简单环即不经过重复的结点的环。

给一个n，对于每一个$1 \le i \le n$，求i个点的不同的有标号仙人掌的个数

$n \le 30000$

仙人掌——解答

设n个点的有根仙人掌个数为c_n，对应的指数生成函数为$C(x)$。

仙人掌——解答

设n个点的有根仙人掌个数为c_n，对应的指数生成函数为$C(x)$。

考虑与根相邻的每一条独立边与每一个环，一条独立边对应的指数生成函数是$C(x)$，一个$i+1$个点的环对应的指数生成函数是$\frac{C(x)^i}{2}$。环和独立边可以自由排列，因此与根相邻的所有环和独立边对应的指数生成函数是$\exp(C(x)+\sum_{n\geq 2}\frac{C(x)^n}{2})=\exp\frac{2C(x)-C(x)^2}{2-2C(x)}$。

仙人掌——解答

设n个点的有根仙人掌个数为c_n，对应的指数生成函数为$C(x)$。

考虑与根相邻的每一条独立边与每一个环，一条独立边对应的指数生成函数是$C(x)$，一个$i+1$个点的环对应的指数生成函数是$\frac{C(x)^i}{2}$。环和独立边可以自由排列，因此与根相邻的所有环和独立边对应的指数生成函数是$\exp(C(x)+\sum_{n\geq 2}\frac{C(x)^n}{2})=\exp\frac{2C(x)-C(x)^2}{2-2C(x)}$。

把根加回去得到方程$C(x)=x\exp\frac{2C(x)-C(x)^2}{2-2C(x)}$，使用牛顿迭代解即可。

时间复杂度$O(n\log n)$。

点双连通图

如果一个无向连通图删去任意一个点都仍然是连通的，我们就称之为点双连通图。

点双连通图

如果一个无向连通图删去任意一个点都仍然是连通的，我们就称之为点双连通图。

给一个n，求n个点的不同的有标号点双连通图的个数

点双连通图

如果一个无向连通图删去任意一个点都仍然是连通的，我们就称之为点双连通图。

给一个n，求n个点的不同的有标号点双连通图的个数

$n \leq 30000$

点双连通图——解答

设n个点的点双连通图个数为b_n，n个点的有根连通图个数为d_n，对应的指数生成函数分别为$B(x), D(x)$。

点双连通图——解答

设n个点的点双连通图个数为b_n，n个点的有根连通图个数为d_n，对应的指数生成函数分别为$B(x), D(x)$。

考虑包含根的点双连通分量，一个$i+1$个点的点双连通分量对应的生成函数是$\frac{b_{i+1}D(x)^i}{i!}$。点双连通分量可以自由排列，因此包含根的所有点双连通分量对应的生成函数是$\exp\sum_{n\geq 1}\frac{b_{n+1}D(x)^n}{n!}=\exp B'(D(x))$。

点双连通图——解答

设n个点的点双连通图个数为b_n，n个点的有根连通图个数为d_n，对应的指数生成函数分别为$B(x), D(x)$。

考虑包含根的点双连通分量，一个$i+1$个点的点双连通分量对应的生成函数是$\frac{b_{i+1}D(x)^i}{i!}$。点双连通分量可以自由排列，因此包含根的所有点双连通分量对应的生成函数是$\exp\sum_{n\geq 1}\frac{b_{n+1}D(x)^n}{n!}=\exp B'(D(x))$。

把根加回去得到方程$D(x)=x\exp B'(D(x))$，并不能使用牛顿迭代解。

点双连通图——解答

设n个点的点双连通图个数为b_n，n个点的有根连通图个数为d_n，对应的指数生成函数分别为$B(x), D(x)$。

考虑包含根的点双连通分量，一个$i+1$个点的点双连通分量对应的生成函数是$\frac{b_{i+1}D(x)^i}{i!}$。点双连通分量可以自由排列，因此包含根的所有点双连通分量对应的生成函数是$\exp\sum_{n\geq 1}\frac{b_{n+1}D(x)^n}{n!}=\exp B'(D(x))$。

把根加回去得到方程$D(x)=x\exp B'(D(x))$，并不能使用牛顿迭代解。

设$D(x)$的复合逆为$D^{-1}(x)$，则方程化为$B'(x)=\ln\frac{x}{D^{-1}(x)}$，使用扩展形式的拉格朗日反演公式即可解决。

点双连通图——解答

设n个点的点双连通图个数为b_n，n个点的有根连通图个数为d_n，对应的指数生成函数分别为$B(x), D(x)$。

考虑包含根的点双连通分量，一个$i+1$个点的点双连通分量对应的生成函数是$\frac{b_{i+1}D(x)^i}{i!}$。点双连通分量可以自由排列，因此包含根的所有点双连通分量对应的生成函数是$\exp\sum_{n\geq 1}\frac{b_{n+1}D(x)^n}{n!} = \exp B'(D(x))$。

把根加回去得到方程$D(x) = x\exp B'(D(x))$，并不能使用牛顿迭代解。

设$D(x)$的复合逆为$D^{-1}(x)$，则方程化为$B'(x) = \ln\frac{x}{D^{-1}(x)}$，使用扩展形式的拉格朗日反演公式即可解决。

时间复杂度$O(n\log n)$。

边双连通图

如果一个无向连通图删去任意一条边都仍然是连通的，我们就称之为边双连通图。

边双连通图

如果一个无向连通图删去任意一条边都仍然是连通的，我们就称之为边双连通图。

给一个n，求n个点的不同的有标号边双连通图的个数

边双连通图

如果一个无向连通图删去任意一条边都仍然是连通的，我们就称之为边双连通图。

给一个n，求n个点的不同的有标号边双连通图的个数

$n \leq 30000$

边双连通图——解答

设n个点的有根边双连通图个数为b_n，n个点的有根连通图个数为d_n，对应的指数生成函数分别为$B(x), D(x)$。

边双连通图——解答

设n个点的有根边双连通图个数为b_n，n个点的有根连通图个数为d_n，对应的指数生成函数分别为$B(x), D(x)$。

假设根所在的边双连通分量有n个点，那么边双连通分量对应的指数生成函数为$\frac{b_n x^n}{n!}$。与边双连通分量相邻的一条边对应的指数生成函数为$nD(x)$。边可以自由排列，因此与边双连通分量相邻的所有边对应的指数生成函数为$\exp(nD(x))$。

边双连通图——解答

设n个点的有根边双连通图个数为b_n，n个点的有根连通图个数为d_n，对应的指数生成函数分别为$B(x), D(x)$。

假设根所在的边双连通分量有n个点，那么边双连通分量对应的指数生成函数为$\frac{b_n x^n}{n!}$。与边双连通分量相邻的一条边对应的指数生成函数为$nD(x)$。边可以自由排列，因此与边双连通分量相邻的所有边对应的指数生成函数为$\exp(nD(x))$。

乘起来后对n累加可以得到方程$D(x) = \sum_{n\geq 1} \frac{b_n x^n \exp(nD(x))}{n!}$，化简得到$D(x) = B(x \exp D(x))$，并不能使用牛顿迭代解。

边双连通图——解答

设n个点的有根边双连通图个数为b_n，n个点的有根连通图个数为d_n，对应的指数生成函数分别为$B(x), D(x)$。

假设根所在的边双连通分量有n个点，那么边双连通分量对应的指数生成函数为$\frac{b_n x^n}{n!}$。与边双连通分量相邻的一条边对应的指数生成函数为$nD(x)$。边可以自由排列，因此与边双连通分量相邻的所有边对应的指数生成函数为$\exp(nD(x))$。

乘起来后对n累加可以得到方程$D(x) = \sum_{n\geq 1} \frac{b_n x^n \exp(nD(x))}{n!}$，化简得到$D(x) = B(x \exp D(x))$，并不能使用牛顿迭代解。

设$x \exp D(x)$的复合逆为$D^{-1}(x)$，则方程化为$B(x) = D(D^{-1}(x))$，使用扩展形式的拉格朗日反演公式即可解决。

边双连通图——解答

设n个点的有根边双连通图个数为b_n，n个点的有根连通图个数为d_n，对应的指数生成函数分别为$B(x), D(x)$。

假设根所在的边双连通分量有n个点，那么边双连通分量对应的指数生成函数为$\frac{b_n x^n}{n!}$。与边双连通分量相邻的一条边对应的指数生成函数为$nD(x)$。边可以自由排列，因此与边双连通分量相邻的所有边对应的指数生成函数为$\exp(nD(x))$。

乘起来后对n累加可以得到方程$D(x) = \sum_{n\ge1} \frac{b_n x^n \exp(nD(x))}{n!}$，化简得到$D(x) = B(x \exp D(x))$，并不能使用牛顿迭代解。

设$x \exp D(x)$的复合逆为$D^{-1}(x)$，则方程化为$B(x) = D(D^{-1}(x))$，使用扩展形式的拉格朗日反演公式即可解决。

时间复杂度$O(n \log n)$。

A+B problem

定义有向图的权值为：将有向图中的每一个强连通分量缩成点，得到的新图中入度为0的点数。

A+B problem

定义有向图的权值为：将有向图中的每一个强连通分量缩成点，得到的新图中入度为0的点数。

给出一个数集s和一个正整数n，定义有向图是合法的当且仅当每个强连通分量大小都在s内。对于每一个$i(1 \leq i \leq n)$，求所有点数为i 的合法有向图的权值的期望。

A+B problem

定义有向图的权值为：将有向图中的每一个强连通分量缩成点，得到的新图中入度为0的点数。

给出一个数集s和一个正整数n，定义有向图是合法的当且仅当每个强连通分量大小都在s内。对于每一个$i(1 \le i \le n)$，求所有点数为i 的合法有向图的权值的期望。

所有数字均不超过10^5。

有向无环图计数

设n个点有向无环图个数为g_n，$A_{n,i}$表示所有点数为n且点i入度为0的有向无环图组成的集合，则$g_n = |\bigcup_{i=1}^{n} A_{n,i}|$。根据容斥原理有：

有向无环图计数

设n个点有向无环图个数为g_n，$A_{n,i}$表示所有点数为n且点i入度为0的有向无环图组成的集合，则$g_n = |\bigcup_{i=1}^{n} A_{n,i}|$。根据容斥原理有：

$$|\bigcup_{i=1}^{n} A_{n,i}| = \sum_{k=1}^{n} (-1)^{k-1} \sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} |\bigcap_{j=1}^{k} A_{n,i_j}|$$

有向无环图计数

设n个点有向无环图个数为g_n，$A_{n,i}$表示所有点数为n且点i入度为0的有向无环图组成的集合，则$g_n = |\bigcup_{i=1}^{n} A_{n,i}|$。根据容斥原理有：

$$|\bigcup_{i=1}^{n} A_{n,i}| = \sum_{k=1}^{n} (-1)^{k-1} \sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} |\bigcap_{j=1}^{k} A_{n,i_j}|$$

注意对于任意一个k，每一组$i_1, i_2, ..., i_k$的$|\bigcap_{j=1}^{k} A_{n,i_j}|$都是相等的，把这个值记作$a_{n,k}$。显然把这$k$个点删掉之后，得到的仍然是一张有向无环图。考虑这$k$个点与其他$n-k$个点之间的每一条边是否存在，即可算出$a_{n,k} = g_{n-k} * 2^{k(n-k)}$。$i_1, i_2, ..., i_k$ 一共有$\binom{n}{k}$组，于是我们可以写出一个等式：

有向无环图计数

设n个点有向无环图个数为g_n，$A_{n,i}$表示所有点数为n且点i入度为0的有向无环图组成的集合，则$g_n = |\bigcup_{i=1}^{n} A_{n,i}|$。根据容斥原理有：

$$|\bigcup_{i=1}^{n} A_{n,i}| = \sum_{k=1}^{n} (-1)^{k-1} \sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} |\bigcap_{j=1}^{k} A_{n,i_j}|$$

注意对于任意一个k，每一组$i_1, i_2, \ldots, i_k$的$|\bigcap_{j=1}^{k} A_{n,i_j}|$都是相等的，把这个值记作$a_{n,k}$。显然把这$k$个点删掉之后，得到的仍然是一张有向无环图。考虑这$k$个点与其他$n-k$个点之间的每一条边是否存在，即可算出$a_{n,k} = g_{n-k} * 2^{k(n-k)}$。$i_1, i_2, \ldots, i_k$ 一共有$\binom{n}{k}$组，于是我们可以写出一个等式：

$$g_0 = 1, g_n = \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} g_{n-k} 2^{k(n-k)}$$

强连通图计数

设h_n表示n个点的有向图个数，d_n表示n个点的强连通图个数。显然$h_n = 2^{n(n-1)}$，考虑如何根据h_n计算出d_n。

强连通图计数

设h_n表示n个点的有向图个数，d_n表示n个点的强连通图个数。显然$h_n = 2^{n(n-1)}$，考虑如何根据h_n计算出d_n。

将强连通分量缩成点，得到的是一张有向无环图。但是有向无环图保证了每一个强连通分量都只有1 个点，现在没有这条性质了，所以我们需要对原式进行修正。

强连通图计数

设h_n表示n个点的有向图个数，d_n表示n个点的强连通图个数。显然$h_n = 2^{n(n-1)}$，考虑如何根据h_n计算出d_n。

将强连通分量缩成点，得到的是一张有向无环图。但是有向无环图保证了每一个强连通分量都只有1 个点，现在没有这条性质了，所以我们需要对原式进行修正。

原式中$(-1)^{k-1}$中的k指的是强连通分量个数，而$\binom{n}{k}g_{n-k}2^{k(n-k)}$中的$k$指的是点的个数，所以需要将$(-1)^{k-1}$替换掉。

强连通图计数——续

定义集合S的权值为$(-1)^{|S|}$，设e_n表示所有元素是强连通图的集合中，集合内元素总点数为n的集合的权值和。稍加推导可以得出新的递推式：

强连通图计数——续

定义集合S的权值为$(-1)^{|S|}$，设e_n表示所有元素是强连通图的集合中，集合内元素总点数为n的集合的权值和。稍加推导可以得出新的递推式：

$$e_n = -\sum_{i=1}^{n} e_{n-i} * h_i * 2^{i(n-i)} * \binom{n}{i}$$

强连通图计数——续

定义集合S的权值为$(-1)^{|S|}$，设e_n表示所有元素是强连通图的集合中，集合内元素总点数为n的集合的权值和。稍加推导可以得出新的递推式：

$$e_n = -\sum_{i=1}^{n} e_{n-i} * h_i * 2^{i(n-i)} * \binom{n}{i}$$

接下来考虑如何根据e_n计算d_n。把点1所在的强连通图从集合中删去，得到的仍然是一个元素是强连通图的集合。于是写出递推式：

强连通图计数——续

定义集合S的权值为$(-1)^{|S|}$，设e_n表示所有元素是强连通图的集合中，集合内元素总点数为n的集合的权值和。稍加推导可以得出新的递推式：

$$e_n = -\sum_{i=1}^{n} e_{n-i} * h_i * 2^{i(n-i)} * \binom{n}{i}$$

接下来考虑如何根据e_n计算d_n。把点1所在的强连通图从集合中删去，得到的仍然是一个元素是强连通图的集合。于是写出递推式：

$$d_n = -e_n - \sum_{i=1}^{n-1} d_i * e_{n-i} * \binom{n-1}{i-1}$$

A+B problem——解答

得到了d_n，便可以反过来得到合法的强连通图个数d'_n，所有元素是合法强连通图的集合中，集合内元素总点数为n的集合的权值和e'_n和n个点的合法有向图个数h'_n。

A+B problem——解答

得到了d_n，便可以反过来得到合法的强连通图个数d'_n，所有元素是合法强连通图的集合中，集合内元素总点数为n的集合的权值和e'_n和n个点的合法有向图个数h'_n。

接下来考虑如何计算所有n个点的合法有向图的权值和ans_n。考虑每个合法强连通分量对答案产生的贡献可以得到：

A+B problem——解答

得到了d_n，便可以反过来得到合法的强连通图个数d'_n，所有元素是合法强连通图的集合中，集合内元素总点数为n的集合的权值和e'_n和n个点的合法有向图个数h'_n。

接下来考虑如何计算所有n个点的合法有向图的权值和ans_n。考虑每个合法强连通分量对答案产生的贡献可以得到：

$$ans_n = \sum_{i=1}^{n} d'_i * h'_{n-i} * 2^{i(n-i)} * \binom{n}{i}$$

A+B problem——解答

得到了d_n，便可以反过来得到合法的强连通图个数d'_n，所有元素是合法强连通图的集合中，集合内元素总点数为n的集合的权值和e'_n和n个点的合法有向图个数h'_n。

接下来考虑如何计算所有n个点的合法有向图的权值和ans_n。考虑每个合法强连通分量对答案产生的贡献可以得到：

$$ans_n = \sum_{i=1}^{n} d'_i * h'_{n-i} * 2^{i(n-i)} * \binom{n}{i}$$

A+B problem——解答——续

定义一种一元运算符Δ。设$F(x)=\sum_{i\geq 0}\frac{f_i * x^i}{i!}$，定义$\Delta F(x)=\sum_{i\geq 0}\frac{f_i * x^i}{i! * 2^{\binom{i}{2}}}$。即$\Delta$是将指数生成函数转化成组合生成函数。

A+B problem——解答——续

定义一种一元运算符Δ。设$F(x)=\sum_{i\geq 0}\frac{f_i*x^i}{i!}$，定义$\Delta F(x)=\sum_{i\geq 0}\frac{f_i*x^i}{i!*2^{\binom{i}{2}}}$。即Δ是将指数生成函数转化成组合生成函数。

设$h_n, e_n, d_n, d'_n, e'_n, h'_n, ans_n$对应的指数生成函数分别为$H(x)$,$E(x)$,$D(x)$,$D_1(x)$,$E_1(x)$,$H_1(x)$,$A(x)$，则之前的递推式转化如下：

A+B problem——解答——续

定义一种一元运算符Δ。设$F(x)=\sum_{i\geq 0}\frac{f_i*x^i}{i!}$，定义$\Delta F(x)=\sum_{i\geq 0}\frac{f_i*x^i}{i!*2^{\binom{i}{2}}}$。即Δ是将指数生成函数转化成组合生成函数。

设$h_n, e_n, d_n, d'_n, e'_n, h'_n, ans_n$对应的指数生成函数分别为$H(x)$,$E(x)$,$D(x)$,$D_1(x)$,$E_1(x)$,$H_1(x)$,$A(x)$，则之前的递推式转化如下：

$$\Delta E(x)=(\Delta H(x))^{-1}$$

$$D(x)=-\ln E(x)$$

$$d'_i=d_i*s_i$$

$$E_1(x)=\mathrm{e}^{-D_1(x)}$$

$$\Delta H_1(x)=(\Delta E_1(x))^{-1}$$

$$\Delta A(x)=\Delta H_1(x)\Delta D_1(x)$$

A+B problem——解答——续

定义一种一元运算符Δ。设$F(x)=\sum_{i\geq 0}\frac{f_i*x^i}{i!}$，定义$\Delta F(x)=\sum_{i\geq 0}\frac{f_i*x^i}{i!*2^{\binom{i}{2}}}$。即$\Delta$是将指数生成函数转化成组合生成函数。

设$h_n, e_n, d_n, d'_n, e'_n, h'_n, ans_n$对应的指数生成函数分别为$H(x)$,$E(x)$,$D(x)$,$D_1(x)$,$E_1(x)$,$H_1(x)$,$A(x)$，则之前的递推式转化如下：

$$\Delta E(x)=(\Delta H(x))^{-1}$$

$$D(x)=-\ln E(x)$$

$$d'_i=d_i*s_i$$

$$E_1(x)=\mathrm{e}^{-D_1(x)}$$

$$\Delta H_1(x)=(\Delta E_1(x))^{-1}$$

$$\Delta A(x)=\Delta H_1(x)\Delta D_1(x)$$

时间复杂度$O(n\log n)$。

淘汰赛

一棵树是合法的当且仅当每个非叶结点的孩子数量都> 1。

淘汰赛

一棵树是合法的当且仅当每个非叶结点的孩子数量都>1。

给出正整数n和m，询问有多少棵不同的含有n个叶结点且每个叶结点深度都为m的树(根的深度为0)。

淘汰赛

一棵树是合法的当且仅当每个非叶结点的孩子数量都> 1。

给出正整数n和m，询问有多少棵不同的含有n个叶结点且每个叶结点深度都为m的树(根的深度为0)。

两棵树是相同的当且仅当结构相同(孩子之间有顺序)。

淘汰赛

一棵树是合法的当且仅当每个非叶结点的孩子数量都> 1。

给出正整数n和m，询问有多少棵不同的含有n个叶结点且每个叶结点深度都为m的树(根的深度为0)。

两棵树是相同的当且仅当结构相同(孩子之间有顺序)。

$1 \leq m \leq 15, n \leq 10^{18}$。

淘汰赛——解答

设$f_{m,n}$表示含有n个叶结点且每个叶结点深度都为m的树的个数，对应的普通生成函数为$F_m(x)$。

淘汰赛——解答

设$f_{m,n}$表示含有n个叶结点且每个叶结点深度都为m的树的个数，对应的普通生成函数为$F_m(x)$。

$m=0$时$F_0(x)=x$

淘汰赛——解答

设$f_{m,n}$表示含有n个叶结点且每个叶结点深度都为m的树的个数，对应的普通生成函数为$F_m(x)$。

$m=0$时$F_0(x)=x$

枚举根的孩子个数可以得到$F_m(x)=\sum_{n\geq 2}F_{m-1}(x)^n=\frac{F_{m-1}(x)^2}{1-F_{m-1}(x)}$

淘汰赛——解答

设$f_{m,n}$表示含有n个叶结点且每个叶结点深度都为m的树的个数，对应的普通生成函数为$F_m(x)$。

$m = 0$时$F_0(x) = x$

枚举根的孩子个数可以得到$F_m(x) = \sum_{n\geq 2} F_{m-1}(x)^n = \frac{F_{m-1}(x)^2}{1-F_{m-1}(x)}$

设$F_m(x) = \frac{P_m(x)}{Q_m(x)}$，则$P_m(x) = P_{m-1}(x)^2, Q_m(x) = Q_{m-1}(x)$
$(Q_{m-1}(x) - P_{m-1}(x))$

淘汰赛——解答

设$f_{m,n}$表示含有n个叶结点且每个叶结点深度都为m的树的个数，对应的普通生成函数为$F_m(x)$。

$m = 0$时$F_0(x) = x$

枚举根的孩子个数可以得到$F_m(x) = \sum_{n\geq 2} F_{m-1}(x)^n = \frac{F_{m-1}(x)^2}{1-F_{m-1}(x)}$

设$F_m(x) = \frac{P_m(x)}{Q_m(x)}$，则$P_m(x) = P_{m-1}(x)^2, Q_m(x) = Q_{m-1}(x)(Q_{m-1}(x) - P_{m-1}(x))$

$P_m(x) = x^{2^m}$，可以在$O(m2^m)$的时间复杂度内算出$Q_m(x)$。

淘汰赛——解答

设$f_{m,n}$表示含有n个叶结点且每个叶结点深度都为m的树的个数，对应的普通生成函数为$F_m(x)$。

$m=0$时$F_0(x)=x$

枚举根的孩子个数可以得到$F_m(x)=\sum_{n\geq 2}F_{m-1}(x)^n=\frac{F_{m-1}(x)^2}{1-F_{m-1}(x)}$

设$F_m(x)=\frac{P_m(x)}{Q_m(x)}$，则$P_m(x)=P_{m-1}(x)^2, Q_m(x)=Q_{m-1}(x)(Q_{m-1}(x)-P_{m-1}(x))$

$P_m(x)=x^{2^m}$，可以在$O(m2^m)$的时间复杂度内算出$Q_m(x)$。

于是只需要求出$[x^{n-2^m}]\frac{1}{Q_m(x)}$即可。

淘汰赛——解答

设$f_{m,n}$表示含有n个叶结点且每个叶结点深度都为m的树的个数，对应的普通生成函数为$F_m(x)$。

$m = 0$时$F_0(x) = x$

枚举根的孩子个数可以得到$F_m(x) = \sum_{n\geq 2} F_{m-1}(x)^n = \frac{F_{m-1}(x)^2}{1-F_{m-1}(x)}$

设$F_m(x) = \frac{P_m(x)}{Q_m(x)}$，则$P_m(x) = P_{m-1}(x)^2, Q_m(x) = Q_{m-1}(x)(Q_{m-1}(x) - P_{m-1}(x))$

$P_m(x) = x^{2^m}$，可以在$O(m2^m)$的时间复杂度内算出$Q_m(x)$。

于是只需要求出$[x^{n-2^m}]\frac{1}{Q_m(x)}$即可。

如果n是10^5那么多项式求逆即可，但现在n是10^{18}。

淘汰赛——解答——续

设$Q_m(x) = 1 + \sum_{n=1}^{2^m-1} q_{m,n}x^n$, $\frac{1}{Q_m(x)} = \sum_{n\geq 0} a_{m,n}x^n$

淘汰赛——解答——续

设$Q_m(x) = 1 + \sum_{n=1}^{2^m-1} q_{m,n}x^n$, $\frac{1}{Q_m(x)} = \sum_{n\geq 0} a_{m,n}x^n$

$n \geq 2^m$时$a_{m,n} + \sum_{i=1}^{2^m-1} q_{m,i}a_{m,n-i} = [x^n](Q_m(x)\frac{1}{Q_m(x)}) = [x^n]1 = 0$

淘汰赛——解答——续

设$Q_m(x) = 1 + \sum_{n=1}^{2^m-1} q_{m,n}x^n$, $\frac{1}{Q_m(x)} = \sum_{n\geq 0} a_{m,n}x^n$

$n \geq 2^m$时$a_{m,n} + \sum_{i=1}^{2^m-1} q_{m,i}a_{m,n-i} = [x^n](Q_m(x)\frac{1}{Q_m(x)}) = [x^n]1 = 0$

即$a_{m,n}$满足常系数线性递推关系。

淘汰赛——解答——续

设$Q_m(x) = 1 + \sum_{n=1}^{2^m-1} q_{m,n}x^n$, $\frac{1}{Q_m(x)} = \sum_{n\geq 0} a_{m,n}x^n$
$n \geq 2^m$时$a_{m,n} + \sum_{i=1}^{2^m-1} q_{m,i}a_{m,n-i} = [x^n](Q_m(x)\frac{1}{Q_m(x)}) = [x^n]1 = 0$
即$a_{m,n}$满足常系数线性递推关系。

一般线性递推式是使用矩阵乘法计算的，可以使用特征多项式优化。(具体原理略去)

淘汰赛——解答——续

设$Q_m(x) = 1 + \sum_{n=1}^{2^m-1} q_{m,n}x^n$, $\frac{1}{Q_m(x)} = \sum_{n\geq 0} a_{m,n}x^n$

$n \geq 2^m$时$a_{m,n} + \sum_{i=1}^{2^m-1} q_{m,i}a_{m,n-i} = [x^n](Q_m(x)\frac{1}{Q_m(x)}) = [x^n]1 = 0$

即$a_{m,n}$满足常系数线性递推关系。

一般线性递推式是使用矩阵乘法计算的，可以使用特征多项式优化。(具体原理略去)

本题矩阵的特征多项式是$Q_m(x)$。设$G(x) = x^{n-2^m} \mod Q_m(x)$，则$a_{m,n-2^m} = \sum_{i=0}^{2^m-2} a_{m,i}[x^i]G(x)$。

$a_{m,i}$可以通过多项式求逆得到，$G(x)$也可以通过快速幂+多项式带余除法得到，因此时间复杂度为$O(m2^m \log n)$。

The end

祝大家明天考试顺利！