计数相关知识

ditoly

组合数

- C_n^m表示从n个不同元素中取出m个元素的方案数
- $C_n^m = C_{n-1}^m + C_{n-1}^{m-1}$
- $C_n^m = \frac{n!}{(n-m)!m!}$

NOIP2016 组合数问题

- 给定k，有t次询问，每次给出n, m，求有多少个整数对(i, j)满足$0 \le i \le n$，$0 \le j \le \min(i, m)$，且C_i^j是k的倍数。
- $n, m \le 2000$，$k \le 21$，$t \le 10^4$

NOIP2016 组合数问题

- C_i^j是k的倍数当且仅当$C_i^j \ mod \ k = 0$
- 用$C_n^m = C_{n-1}^m + C_{n-1}^{m-1}$随便算算即可

SHOI2017 组合数问题

- 给定四个整数n, p, k, r，求$\sum_{i=0}^{\infty} C_{nk}^{ik+r} \ mod \ p$
- $1 \le n \le 10^9$，$0 \le r < k \le 50$，$2 \le p \le 2^{30} - 1$

SHOI2017 组合数问题

- $f(i,j)$表示前i个数选了x个数，且$x \ mod \ k = j$的方案数
- $f(i,j) = f(i-1,j) + f(i-1,(j-1) mod \ k)$
- 矩阵乘法优化即可

清华集训2016 组合数问题

- 给定k，有t次询问，每次给出n, m，求有多少个整数对(i, j)满足$0 \le i \le n$，$0 \le j \le \min(i, m)$，且C_i^j是k的倍数。
- $n, m \le 10^{18}$，$1 \le t, k \le 100$，k为质数

$lucas$定理

- $C_{ax+b}^{cx+d} \equiv C_a^c \cdot C_b^d \ (mod\ x)$
- $C_n^m \equiv C_{\left\lfloor \frac{n}{x} \right\rfloor}^{\left\lfloor \frac{m}{x} \right\rfloor} \cdot C_{n\ mod\ x}^{m\ mod\ x} \ (mod\ x)$
- x为质数

清华集训2016 组合数问题

- 根据$lucas$定理，C_i^j为k的倍数的条件是k进制下，i有某一位小于j
- 数位DP随便做一下就好了

CTSC2017 吉夫特

- 输入一个长度为n的数列a_i
- 求有多少个长度大等于2的不上升子序列$a_{b_1}, a_{b_2}, \ldots, a_{b_k}$满足$\prod_{i=2}^{k} C_{a_{b_{i-1}}}^{a_{b_i}} \ mod\ 2 > 0$。
- $1 \le n \le 211985$，$1 \le a_i \le 233333$，所有a_i互不相同

CTSC2017 吉夫特

- 乘积模2大于0相当于每个$C_{a_{b_{i-1}}}^{a_{b_i}}$模2都非0
- 根据$lucas$定理，条件可以转化为$a_{b_{i-1}}$二进制下每一位都不小于a_{b_i}对应的二进制位
- 用$f(i)$表示最后一个元素为i的合法子序列个数
- 依次转移每个$f(a_i)$，直接枚举a_i的超集转移，令$N=\max\{\log a_i\}$，时间复杂度为$O(3^N)$。

CTSC2017 吉夫特

- 还可以分二进制下前$\frac{N}{2}$位和后$\frac{N}{2}$位，维护数组$s(x)$，表示前$\frac{N}{2}$位与x相同，后$\frac{N}{2}$位是x的超集的i对应的$f(i)$之和。
- 转移时枚举前$\frac{N}{2}$位的超集转移，再枚举后$\frac{N}{2}$位的子集维护$s(x)$，总时间复杂度$O(6^{\frac{N}{2}})$。

计算组合数

- 预处理阶乘和逆元
- $O(m)$暴力算
- $O(nm)$预处理
- 分解质因数
- $lucas$定理
- 中国剩余定理

计算组合数

- 求$C_n^m \ mod \ 10^9+7$
- $n,m \le 10^9$

计算组合数

- 转化为求阶乘
- 设值K，构建多项式$F(x) = \prod_{i=1}^{K}(x+i)$
- $n! = \prod_{i=0}^{\left\lfloor \frac{n}{K} \right\rfloor - 1} F(iK) \cdot \prod_{i=k\left\lfloor \frac{n}{k} \right\rfloor + 1}^{n} i$
- 用分治FFT等操作求出$F(x)$，再用多点求值求出所有$F(iK)$
- 时间复杂度$O(\sqrt{n}\log^2 n)$

计算组合数

- 有没有更快更好写的？
- 分块打表

AGC001E BBQ Hard

- 有n组烧烤材料，每组有a_i片牛肉和b_i片青椒
- 某人准备从n组中选出两组材料，将两组中所有牛肉和青椒从左到右串成一串做烧烤
- 求有多少种不同的烧烤方式，两种方式不同当且仅当选的材料不同或者串材料的顺序不同，所有牛肉之间不作区分，所有青椒之间也不作区分
- $2 \le n \le 200000$，$1 \le a_i, b_i \le 2000$

AGC001E BBQ Hard

- 假设选第i组和第j组材料，方案数为$C_{a_i+a_j+b_i+b_j}^{a_i+a_j}$
- C_{n+m}^{n}相当于每次可以向上或向右走一格，从$(0,0)$走到(n,m)的方案数
- $C_{a_i+a_j+b_i+b_j}^{a_i+a_j}$可以看成从$(-a_i,-b_i)$走到$(a_j,b_j)$的方案数
- 以所有$(-a_i,-b_i)$作为初始状态，DP出走到每一格的总方案数，再扣掉多算的部分即可，时间复杂度$O(2000^2)$

简单树题

- 给出一棵n个点的树，对于每个$1 \le k \le n$，求有多少个k个点的集合可以用至多一条路径覆盖
- $1 \le n \le 10^5$

简单树题

- $k = 1$时，答案显然为n
- $k > 1$时，考虑在路径最边上的两个点，那么对于树上每条有l个点的路径，对答案的贡献为C_{l-2}^{k-2}
- 点分治+FFT对每个l求出有多少条l个点的路径
- 组合数拆成阶乘可以表示成卷积的形式，再做一遍FFT即可
- 时间复杂度$O(n\log^2 n)$

生成函数

- 数列a_i的普通生成函数为$\sum_{i=0}^{\infty} a_i x^i$
- 一种理解：a_i表示选i个物品的方案数，每个x代表一个物品
- 如果能拆分出若干个互不相关的选择，每个选择的生成函数的乘积就是总方案数的生成函数
- $(x+1)^n = \sum_{i=0}^{n} C_n^i x^i$
- $(x+y)^n = \sum_{i=0}^{n} C_n^i x^i y^{n-i}$

BZOJ 3771 Triple

- 有若干个互不相同的正整数a_i，从中选出至多三个，求选出的数的和为每种情况的方案数
- $a_i \leq 40000$

BZOJ 3771 Triple

- 把值看成“物品”，和每增加1，看成选了一个“物品”
- 选出一个数的所有方案的生成函数显然为$\sum x^{a_i}$
- 假设可以重复选同一个数，并且选的物品之间有第几次选的区别，那么依次选3个物品，3次都是独立的选择，生成函数为$(\sum x^{a_i})^3$
- 用容斥去掉重复选同一个数的方案，$\sum x^{2a_i}$，$\sum x^{3a_i}$分别为选一个数两次和三次的生成函数，不难求出不重复选的方案数，除以阶乘即可得到答案
- 用FFT做卷积，时间复杂度$O(n\log n)$

简单例题

- n个物品，第i个物品重量为a_i，每个物品可以选任意次，对于每个$1 \le i \le m$，求有多少种方案选出的重量和为i，两种方案不同当且仅当某个物品选的个数不同
- $n, m \le 10^5$

简单例题

- 显然每个物品独立，每个物品的生成函数为$\sum_{k=0}^{\infty} x^{k\cdot a_i}$
- 直接卷积复杂度太大，考虑求ln 后加起来再exp
- $B(x) = \ln A(x)$
- $B'(x) = \frac{A\prime(x)}{A(x)}$
- $\sum_{k=0}^{\infty} x^{k\cdot a_i} = \frac{1}{1\text{-}x^{a_i}}$
- $B'(x) = A'(x) \cdot (1 - x^{a_i})$
- 根据调和级数，总共$O(m\log m)$项，总时间复杂度$O(m\log m)$

第一类斯特林数

- $S_1(n,k)$表示把n个元素排成k个环的方案数
- $S_1(n,k) = S_1(n-1,k-1) + (n-1) \cdot S_1(n-1,k)$
- 把环看成“物品”，由$n-1$转移到n的过程可以看成有一种方案增加一个物品，$n-1$种方案不增加物品
- 每次转移只与n有关，与k无关，故每次选择独立
- $\sum_{i=0}^{n} S_1(n,i) \cdot x^i = \prod_{i=0}^{n-1}(x+i)$

第一类斯特林数

- 根据生成函数可以分治+FFT用$O(n\log^2 n)$的时间对一个n求出所有$S_1(n,k)$
- 也可以倍增，令$F_n(x)=\prod_{i=0}^{n-1}(x+i)$
- $F_{2n}(x)=F_n(x)\cdot F_n(x+n)$
- $F_n(x+n)=\sum_{i=0}^{n}(x^i\cdot\sum_{j=i}^{n}C_j^i\cdot S_1(n,j)\cdot n^{j-i})$
- 可以表示成卷积形式，倍增+FFT即可，总时间复杂度$O(n\log n)$

第一类斯特林数

- 我们定义上升幂函数和下降幂函数
- $A_n(x) = \prod_{i=0}^{n-1}(x+i)$
- $B_n(x) = \prod_{i=0}^{n-1}(x-i)$
- 第一类斯特林数即上升幂展开式的各项系数
- 乘上$(-1)^{n-k}$即下降幂展开式的各项系数
- 下降幂展开式的系数也被称为有符号第一类斯特林数

一个题

- 有m个正整数变量，求这些变量有多少种取值满足和不超过s且前n个变量均不超过t
- $m - n \leq 1000$，$m \leq 10^9$，$t \leq 10^5$，$nt \leq s \leq 10^{18}$

一个题

- 我们令S表示s减去前n个变量的值
- 确定前n个变量后，剩下的变量方案数为C_S^{m-n}
- 注意到组合数可以表示成下降幂除以阶乘的形式，我们只需要对每个$0 \le k \le m-n$求出前n个变量所有情况下S^k的和，再用第一类斯特林数即可算出答案
- $(S-x)^k = \sum_{i=0}^{k} C_k^i \cdot S^i \cdot (-x)^{k-i}$
- 减去一个值相当于把我们当前求出的所有S^k卷积上一个多项式
- 倍增这个多项式即可，减去的这个数的 k 次方的和即自然数幂和，求法很多，总时间复杂度可以做到$O((m-n)^2)$

第二类斯特林数

- $S_2(n,k)$表示n个元素分成k个非空集合的方案数
- $S_2(n,k)=S_2(n-1,k-1)+k\cdot S_2(n-1,k)$
- 选择之间不是那么独立，不好用生成函数
- 容斥，$S_2(n,k)=\frac{\sum_{i=0}^{k}(-1)^i\cdot C_k^i\cdot(k-i)^n}{k!}$
- 发现是卷积，可以$O(n\log n)$对一个n求出所有$S_2(n,k)$

CF932E Team Work

- 有n个人，你要从中选出一个非空子集，选出x个人的贡献是x^k，求所有方案的贡献之和
- $n \le 10^9$，$k \le 5000$

CF932E Team Work

- 等同于求$\sum_{i=0}^{n} C_n^i \cdot i^k$
- 然而式子优美并不一定有用
- 我们给x^k一个组合意义：我们在被选中的人中进行k次抽奖，每次抽出一个人发奖，同一个人可以被抽中多次
- 我们换一个角度计算这个式子，枚举几个人中奖了
- 答案为$\sum_{i=0}^{k} C_n^i \cdot S_2(k,i) \cdot i! \cdot 2^{n-i}$
- 时间复杂度$O(k^2)$或$O(k\log k)$

生成函数2

- 生成函数的另一种常见用法：解递推式
- 例如，令斐波那契第n项为f_n，则斐波那契数列的生成函数为$F(x)=\sum_{i=0}^{\infty} f_n x^n$
- 根据递推式，构造关于生成函数的等式
- $f_n = f_{n-1} + f_{n-2}$，$f_0 = f_1 = 1$
- $F(x) = F(x)\cdot x + F(x)\cdot x^2 + 1$
- $F(x) = \frac{1}{1-x-x^2}$
- ~~多项式求逆即可求出斐波那契数列的前几项~~

CF438E The Child and Binary Tree

- 给出一个正整数集合c_i，对于所有$1 \le i \le m$求有多少个形态不同的二叉树点权均属于这个集合且点权和为i
- $1 \le m, c_i \le 10^5$

CF438E The Child and Binary Tree

- $f(i)$表示总点权和为i的二叉树数目，$a(x)$表示是否存在$c_i = x$
- $f(0) = 1$，$f(i) = \sum_{x,y} f(x) \cdot f(y) \cdot a(i - x - y)$
- 用$F(x)$表示$f(i)$的生成函数，$A(x)$表示$a(i)$的生成函数
- $F(x) = F(x)^2 \cdot A(x) + 1$
- $F(x) = \frac{1 \pm \sqrt{1 - 4A(x)}}{2A(x)}$
- $A(x)$的常数项是0，用平方和公式化一化
- $F(x) = \frac{2}{1 \pm \sqrt{1 - 4A(x)}}$
- 用分母常数项非0的解就好了，时间复杂度$O(m\log m)$

CF866G Flowers and Chocolate

- 有两类物品，第一类有n种，第i种权值为a_i，第二类有m种，第i种权值为b_i
- 现进行k次选择，每次选出第一类物品中的一种，可以重复选；再进行任意次选择，每次选出第二类物品中的一种，也可以重复选
- 求有多少种方案选出的两类物品的权值和相等
- $1 \le n \le 10$，$1 \le m \le 100$，$1 \le k \le 10^{18}$
- $1 \le a_i \le 10^9$，$1 \le b_i \le 250$

CF866G Flowers and Chocolate

- 显然选第一类物品的生成函数为$(\sum_{i=1}^{n} x^{a_i})^k$
- 假设我们暴力求出了这个生成函数的各项系数，我们只要再进行一次DP即可得到答案，令$f(t)$表示选出的第一类物品和为t的方案数，枚举i转移到$f(t-b_i)$，最后$f(0)$即为答案
- 考虑转移的过程对生成函数的影响，实际上是把$f(t)\cdot x^t$变成了$f(t)\cdot \sum_{i=1}^{m} x^{t-b_i}$
- 于是我们得到等式$x^t=\sum_{i=1}^{m} x^{t-b_i}$
- 直接在模$x^{\max b_i}-\sum_{i=1}^{m} x^{\max\{b_i\}-b_i}$意义下求出生成函数，最后再DP即可得到答案

- 谢谢大家