数学专题题解

A

> 题意：已知方程组 $x_1+x_2+x_3+...+x_n=s$和限制 $0\le x_i\le d_i$

首先当 $x_1,x_2,x_3,...,x_n$ 没有上界限制的时候。

问题转化为了高中的一个经典的组合数学问题，此时答案是

$$\binom{s+n-1}{n-1}$$

如果不知道这个结果是怎么来的话可以搜索 **插板法**

值得一提的是里面的一个代换技巧

令 $x_1'=x_1+1$，于是有$\forall i,x_i'>0$，以及$\sum_{i=1}^n x_i=s-n$

然后将非负整数的问题转化为更好处理的正整数问题了

我们借鉴这个做法可以得到一个容斥的做法。

以两个变量时为例。

我们有 $x_1+x_2=s,x_1\le d_1,x_2\le d_2$

我们定义

$D_1=\{(x,y)|x\ge 0,y\ge 0\}$,

$D_2=\{(x,y)|x\ge d_1+1,y\ge 0\}$,

$D_3=\{(x,y)|x\ge 0,y\ge d_2+1\}$,

$D_4=\{(x,y)|x\ge d_1+1,y\ge d_2+1\}$

四个范围均为其中的整数点

设四个范围内对应解的数量分别为 d_1,d_2,d_3,d_4

那么通过容斥定理或者简单画图我们可以知道这题答案

$$ans = d_1 - d_2 - d_3 + d_4$$

上文以及提到 d_1 怎么求解了，我们考虑 d_2 的值。借鉴上面的思想

我们可以 $x_1' = x_1 - d_1, x_2' = x_2 + 1$

然后就变成我们熟悉的问题了。

20个变量只要简单推广就可以得到答案。

但我其实不是这么做的，容斥的想法有点技巧性，并且我容斥容易写错。

下面简单发现一下我拿到这题的思路

先转化为生成函数的题

相当于求

$$(1 + x + x^2 + ...x^{d_1})(1 + x + x^2 + ...x^{d_2})...(1 + x + x^2 + ...x^{d_n})$$

上面函数的次数为 s 的项的系数

先是等比数列求和

$$\frac{1 - x^{d_1+1}}{1 - x}\frac{1 - x^{d_2+1}}{1 - x}...\frac{1 - x^{d_n+1}}{1 - x}$$

我们先关注分母，我们可以直接记住公式

$$\frac{1}{(1 - x)^n} = (1 + \binom{n}{n-1}x + \binom{n+1}{n-1}x^2 + ... + \binom{n+t-1}{n-1}x^t)$$

而分子只包括2^n个项，直接枚举就行

Q

> 题意：多组询问，每组$n * m$的矩阵,矩阵元素 $a_{i,j} = f_{gcd(i,j)}$，f为斐波那契数列，求矩阵所有元素乘积

首先这题很容易联想到莫比乌斯应用的一道很经典的题

$1 \le x \le n, 1 \le y \le m$

求有多少个 x, y 使得$gcd(x, y) = d$

首先是像 A 题类似的思维令 $x' = dx, y' = dy$

那就相当问范围 $1 \le x \le \lfloor \frac{n}{d} \rfloor, 1 \le y \le \lfloor \frac{m}{d} \rfloor$，有多少个 x, y 满足 $gcd(x, y) = 1$

那下面我们只需要考虑 $d = 1$ 的时候怎么求就行

相当于求

$$\sum_{i=1}^{n}\sum_{j=1}^{m}[gcd(i,j)=1]$$

显然，在没接触过莫比乌斯反演的情况下上面这个式子是相当莫名其妙的

所以，接下来的推导建议自己去 luogu 找 ZAP-Queries做一下，感受一下莫比乌斯怎么用。以及怎么样和整除分块联系在一起的。

下面我给出一些简要推导

$$\begin{aligned} query(n,m) &= \sum_{i=1}^{n}\sum_{j=1}^{m}[gcd(i,j)=1] \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{d|n}\mu(d) \\ &= \sum_{d=1}^{n}\mu(d)\sum_{i=1}^{n}\sum_{j=1}^{m}[i\%d=0][j\%d=0] \\ &= \sum_{d=1}^{n}\mu(d)\lfloor\frac{n}{d}\rfloor\lfloor\frac{m}{d}\rfloor \end{aligned}$$

然后就是整除分块了。

回到本题，联想到该题后一个很自然的想法对每个数 k，求出矩形中有多少个数的是 f_k，假设出现次数的s，那么这个因子对答案的贡献就是f_k^s

那答案相当于

$$\prod_{k=1}^{n} pow(f_k, query(\lfloor\frac{n}{k}\rfloor, \lfloor\frac{m}{k}\rfloor))$$

比较容易观察到，这里也有一个整除分块的存在

同时在不同的多组询问中 $query(n, m)$ 可能被多次调用，我们可以进行记忆化

然后再通过一点卡常的技巧，我们就能够将时间卡到惊人的1920ms，没错，解是非预期的。

T

先把第一个样例的答案丢到OEIS中，得到 $m=2$ 的答案为

$$\sum_{d|n} 2^{\frac{n}{d}}\phi(2d)$$

其中 ϕ 为欧拉函数

然后对照第二个样例，简单枚举可能的答案，我们可以得到下面的公式是同时满足两个样例的

$$\sum_{d|n} m^{\frac{n}{d}}\phi(d)gcd(d,m)$$

然后我们直接对每个 n 求解对应的答案显然是会超时的。我们可以参考交换求和顺序的思路，对每个 d ，求出他对每个答案的贡献

阿巴阿巴，看我今晚能不能学会正解怎么做

怎么这就12点了

好了我学会了。

首先学过**Burnside lemma**的朋友都知道，这就是一道很标准的**Burnside lemma**习题。

我们有元素的集合**C**和一个群**G**，在这里体面，集合C中的元素就是 n 维向量构成的集合 $(x_0,x_1,.....,x_{n-1}),\forall i,0\le x_i\le m$。显然，这个群里面的元素数量是$n^m$个。

群的话是一个置换群。置换群的元素是每个元素都是一个操作，将集合C中的所有元素映射到C中某一元素中，G中的元素可以类似于一个函数 $f:C\to C$。但并不是随便挑几个函数就能构成置换群，他要求对称性。

这题里面其实置换操作可以看成两个操作的符合，一个是旋转 τ ，另一个是全部元素$+k$，我们记为 p ，记 p_k 为变换 $a_i=a_i+k \bmod m$ ，记τ_i 为变换 $a_i=a_{i+k \pmod n}$，那么 $\{\tau_i p_j|i=0,1,2,...n-1,\quad j=0,1,2,...m-1\}$ 可以不重不漏的表示该题所有的等价变换。而且其复合运算构成了一个群。其大小为 nm

此时可能会很有人问为什么会不重不漏？操作 $p_i\tau_j$ （注意操作或者说是函数运算不一定复合交换律）是不是没有被算进去？或者可能根本看不懂上面在讲什么（这也是正常的）。可以先去学一下**抽象代数**

我们回到Burnside lemma。公式是

$$|C/G| = \frac{1}{|G|}\sum_{f\in G}|C(f)|$$

解释符号 $|C/G|$ 表示商群的等价类数量，$|C(f)|$ 是满足 $f(c)=c$ 的 C 中元素数量

我们要求的即为等式左边，$|G|$ 见上文可知是 mn

然后我们就是要计算 $|C(f)|$

我们考虑变换 $\tau_i p_j$ 考虑一下这个不变是什么意思

这个变化就是 $a_k = (a_{k+i \bmod n} + j \pmod m)$

先考虑这个旋转的条件，我们可以将所有不会影响的点分开考虑。比如考虑 $n=12$旋转3下的情况，这样子你显然可以分成 $\{a_0, a_3, a_6, a_9\}, \{a_1, a_4, a_7, a_10\}, \{a_2, a_5, a_8, a_11\}$ 三个部分。我们假设每个部分有 d 个元素，显然 $d|n$,同时分出了 n/d 个部分

置换变化前面没变，也就是说

$$\begin{cases} a_0 \equiv a_3 + j \bmod m \\ a_3 \equiv a_6 + j \bmod m \\ a_6 \equiv a_9 + j \bmod m \\ a_9 \equiv a_0 + j \bmod m \end{cases}$$

那么也就是说要 $4j = 0 \pmod m$

可以观察出来对于一般情况的要求为 $jd \equiv 0 \pmod m$

当 $jd \not\equiv 0 \pmod m$ 的时候，不存在这样的不变元素。

当 $jd \equiv 0 \pmod m$ 的时候，a_0有m种选法，类似的，每一个分组都有 m 种分法，所以总共有 $m^{\frac{n}{d}}$ 种可能

对于给定的 d，我们要找有多少个 j，满足 $0 \le j < m$ 使得 $jd \equiv 0 \pmod m$

也就是 $j|\frac{m}{gcd(m,d)}$，显然有

$$j = \frac{m}{gcd(m,d)}, 2*\frac{m}{gcd(m,d)}, ...gcd(m,d)\frac{m}{gcd(m,d)}$$

这 $gcd(m,d)$ 个解

我们回去考虑旋转，我们现在可以只关心对于每个 d，有多少种旋转使得我们分出来所有部分都有 d 个元素。考虑现在旋转了 i 下，我们可以用类似于并查集或者等价类的方法处理。从 a_0 出发， 走到 a_i (a_0 和 a_i 肯定被划分倒了同一个部分)，再走到 a_{2i}，....直到重新回到 a_0 。那我们也就是找到最小的 j ，使得 $ij \equiv 0 \pmod n$，显然 $j = \frac{n}{gcd(n,i)}$ ，所以此时每个部分有 $\frac{n}{gcd(n,i)}$ 个元素。

此时我们只需要考虑对所有 d，满足 $d|n$ 时能有多少个 i 使得 $gcd(n,i) = n/d$

我们令 $i = i' \frac{n}{d}$ ，那就相当于 $gcd(d,i') = 1$，根据欧拉函数的定义，此时的结果为 $\phi(d)$

考虑枚举 n 个每个因子 d,将上述结果拼接在一起，最终就能得到开头的

$$\sum_{d|n} m^{\frac{n}{d}} \phi(d) gcd(d,m)$$

U

> 题意：$v_1x_1 + v_2x_2 + ... + v_nx_n = d$， $v_1, v_2, ..., v_n$为给定正数，对所有的$d = 1, 2, ...m$，求解方程有多少组非负整数组

先将问题转化为生成函数

因为每样物品都没有没有限制的取，所有就相当于求

$$P = (1 + x^{v_1} + x^{2v_1} + ...)(1 + x^{v_2} + x^{2v_2} + ...)...(1 + x^{v_n} + x^{2v_n} + ...)$$

上面多项式的前 $m+1$ 项

可以先等比数列求和

$$P = \frac{1}{1-x^{v_1}} \frac{1}{1-x^{v_2}} ... \frac{1}{1-x^{v_n}}$$

然后就有很天才的想法，因为这是一个求和式子，所有我们可以有

$$\begin{aligned} Q &= \ln P \\ &= \sum_{i=1}^{n} -\ln(1-x^{v_1}) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{\infty} \frac{x^{v_i j}}{j} (Taylor\text{公式}) \end{aligned}$$

然后我们对每个 v_i 对他求其对多项式造成的影响，这个过程是 $O(n \ln n)$, 涉及调和数列求和，注意当 v_i 全是 1 的时候可能会退化成 $O(n^2)$，可以想想如何避免这种退化

我们要求的是多项式 P，而我们手头有 $Q=\ln P$

我们此时相当于要求 $\exp\{Q\}$ ，首先能很容易想到的思路是泰勒展开，但显然时间会炸。

接着我的思路是对 $Q=\ln P$，两边同时求导。有

$$\frac{P'}{P}=Q'$$

为什么会想到两边求导？因为 P 和 Q 本质还都是多项式，经过求导操作就没有比较奇怪的多项式取对数和求一个数的多项式次方这种奇怪的操作了，只剩比较正统的乘除法。

假设

$$Q'(x)=q_0+q_1x+q_2x^2+...+q_nx^n+...$$

$$P(x)=a_0+a_1x+a_2x^2+...+a_nx^n+...$$

比较显然 $a_0=1$

于是公式

$$P'(x)=P(x)Q'(x)$$

转化为

$$a_1+2a_2x+3a_3x^2+...=(1+a_1x+a_2x^2+...)(q_0+q_1x+q_2x+...)$$

我们简单算一下前几项于是有

$$2a_2=a_1q_0+a_0q_1$$

$$3a_3=a_2q_0+a_1q_1+a_0q_2$$

可以归纳出规律为

$$na_n=\sum_{i=0}^{n-1}a_iq_{n-1-i}$$

然后我们可以很容易想到周六讲座时的半在线卷积。

X

> 题意:给定一棵树,猫猫在两个节点之间漫步,爱因也在两个节点之间走,每两个节点直接距离一样。求是否会相遇,如果会,输出最早相遇的起点。

这题的 n 和 m 比较小, 所以我们可以直接枚举所有猫猫经过的点和所有爱因经过的点,求这些点的交集以及其中每个点的相遇时间，在此中找到最小的即可。

我们可以先只研究猫猫的路径。那我们就可以将一个树上的问题转化为一个铺平了的数列上的问题了。找到这条铺平了的路径可以先找到两个节点的 LCA，然后两个节点一起往上跳到LCA。

我们假设猫猫经过路径长度为 l_1，我们可以对每个经过的点标号 $0, 1, 2, ...l_1$，其中 0 是起点 l_1 是终点，同时 $t = 2l_1$的时候猫猫会走回起始点，开始了循环。那么我们可以得到猫猫经过标号为 i 的点的时间为 $t = i + k2l_1$ 或 $t = (2l_1 - i) + k2l_1$，第二种情况是在回来的路上。

对爱因的路径同理

那对于树上的一个节点 u，假如猫猫经过他的时间是 $t = i_1 + k_1 2l_1$ (另一种情况同理)，而 $t = i_2 + k_2 2l_2$（另一种情况同理）

我们可以转化为同余方程

$$\begin{cases} x \equiv i_1 \pmod{2l_1} \\ x \equiv i_2 \pmod{2l_2} \end{cases}$$

求出每个点的相遇时间，求最小的即可。