$Solution$

作者：JDScript0117

下文只有正解解法，部分分请听讲评

笔者感觉难度应该是 $T3 \le T1 < T2$

$T1$

下文中所有对于复杂度的分析都认为 n, m 同阶

先考虑对于一种固定的情况怎么求 F, W

对于二维矩阵，一个很经典的套路就是将行和列看成点，在第 i 行和第 j 列之间连 $A_{i,j}$ 条边（这个套路的目的是转图论，具象化，有一个性质是一定是二分图）

对于这道题，相当于对于每个 Soldier，将两个可以吃掉它的 Castle 之间连一条边

发现如果存在一个联通块，边数小于点数（树），则 $F = 0$

又因 $F \neq 0$，一共有 $n + m$ 个点却只有 $n + m$ 条边，所以这肯定是一个基环树森林

于是发现 $F = 2^{cnt}$，其中 cnt 可以表示联通块的数量，也可以表示基环树的数量，环的数量等

所以 $W = F^k = \left(2^k\right)^{cnt} = K^{cnt}$，其中 $K = 2^k$ 可以看做一个常数

于是这道题就变成了在 n 个左部点，m 个右部点的完全有编号二分图中生成基环树森林的权值和

发现在 n 个左部点，m 个右部点中一定会各选出 x 个点去组成置换环，且权值只会由置换环中环的个数贡献，定义 f_x 表示 x 个左部点和 x 个右部点组成的置换环的贡献和，有

$$f_0 = 1, f_n = \sum_{i=2}^{n} \binom{n-1}{i-1} \binom{n}{i} \frac{(i-1)!(i)!}{2} K f_{n-i}$$

其中 i 表示枚举第一个左部点所在的环的左右部点个数，这个式子可以 $O(n^2)$ 算出来

那么考虑剩下的 $n - x$ 个左部点和 $m - x$ 个右部点会怎么样，发现它们一定会挂在前面那 x 个左部点和 x 个右部点上，以这 $2x$ 个点作为根建出一个森林

发现这 $2x$ 个点其实可以合并成一个点，于是问题转化为

$n - x$ 个左部点和 $m - x$ 个右部点所形成的完全有编号二分图，与一个和每个点连了 x 条边的特殊点，所构成的图的生成树个数

看到生成树，想到 Matrix Tree 定理，写出矩阵

$$
\left[\begin{array}{c:ccccc:ccccc}
x(n+m-2x) & -x & -x & -x & \cdots & -x & -x & -x & -x & \cdots & -x \\ \hdashline
-x & m & 0 & 0 & \cdots & 0 & -1 & -1 & -1 & \cdots & -1 \\
-x & 0 & m & 0 & \cdots & 0 & -1 & -1 & -1 & \cdots & -1 \\
-x & 0 & 0 & m & \cdots & 0 & -1 & -1 & -1 & \cdots & -1 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\
-x & 0 & 0 & 0 & \cdots & m & -1 & -1 & -1 & \cdots & -1 \\ \hdashline
-x & -1 & -1 & -1 & \cdots & -1 & n & 0 & 0 & \cdots & 0 \\
-x & -1 & -1 & -1 & \cdots & -1 & 0 & n & 0 & \cdots & 0 \\
-x & -1 & -1 & -1 & \cdots & -1 & 0 & 0 & n & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\
-x & -1 & -1 & -1 & \cdots & -1 & 0 & 0 & 0 & \cdots & n
\end{array}\right]
$$

其中第一部分表示一个特殊点，第二部分表示 $n-x$ 个左部点，第三部分表示 $m-x$ 个右部点

定理告诉我们，生成树个数应该为

$$
\left|\begin{array}{ccccc:ccccc}
m & 0 & 0 & \cdots & 0 & -1 & -1 & -1 & \cdots & -1 \\
0 & m & 0 & \cdots & 0 & -1 & -1 & -1 & \cdots & -1 \\
0 & 0 & m & \cdots & 0 & -1 & -1 & -1 & \cdots & -1 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\
0 & 0 & 0 & \cdots & m & -1 & -1 & -1 & \cdots & -1 \\ \hdashline
-1 & -1 & -1 & \cdots & -1 & n & 0 & 0 & \cdots & 0 \\
-1 & -1 & -1 & \cdots & -1 & 0 & n & 0 & \cdots & 0 \\
-1 & -1 & -1 & \cdots & -1 & 0 & 0 & n & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\
-1 & -1 & -1 & \cdots & -1 & 0 & 0 & 0 & \cdots & n
\end{array}\right|
$$

考虑手动消元

用上面 $n-x$ 行很容易将下面 $m-x$ 行的前 $n-x$ 列消掉

$$
\left|\begin{array}{ccccc:ccccc}
m & 0 & 0 & \cdots & 0 & -1 & -1 & -1 & \cdots & -1 \\
0 & m & 0 & \cdots & 0 & -1 & -1 & -1 & \cdots & -1 \\
0 & 0 & m & \cdots & 0 & -1 & -1 & -1 & \cdots & -1 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\
0 & 0 & 0 & \cdots & m & -1 & -1 & -1 & \cdots & -1 \\ \hdashline
0 & 0 & 0 & \cdots & 0 & n-\frac{n-x}{m} & -\frac{n-x}{m} & -\frac{n-x}{m} & \cdots & -\frac{n-x}{m} \\
0 & 0 & 0 & \cdots & 0 & -\frac{n-x}{m} & n-\frac{n-x}{m} & -\frac{n-x}{m} & \cdots & -\frac{n-x}{m} \\
0 & 0 & 0 & \cdots & 0 & -\frac{n-x}{m} & -\frac{n-x}{m} & n-\frac{n-x}{m} & \cdots & -\frac{n-x}{m} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\
0 & 0 & 0 & \cdots & 0 & -\frac{n-x}{m} & -\frac{n-x}{m} & -\frac{n-x}{m} & \cdots & n-\frac{n-x}{m}
\end{array}\right|
$$

于是变成了

$$m^{n-x}\begin{vmatrix}n-\frac{n-x}{m} & -\frac{n-x}{m} & -\frac{n-x}{m} & \cdots & -\frac{n-x}{m}\\ -\frac{n-x}{m} & n-\frac{n-x}{m} & -\frac{n-x}{m} & \cdots & -\frac{n-x}{m}\\ -\frac{n-x}{m} & -\frac{n-x}{m} & n-\frac{n-x}{m} & \cdots & -\frac{n-x}{m}\\ \vdots & \vdots & \vdots & \ddots & \cdots\\ -\frac{n-x}{m} & -\frac{n-x}{m} & -\frac{n-x}{m} & \cdots & n-\frac{n-x}{m}\end{vmatrix}$$

对于最后这个 $(m-x)\times(m-x)$ 的矩阵的行列式，我们考虑行列式的一个运算律

> 对于 $n\times n$ 的矩阵，若 $\forall j\in\{1,2,3,\dots,n\},A_{i,j}=B_{i,j}+C_{i,j}$，有
> $$\begin{vmatrix}A_{1,1} & \cdots & A_{1,i} & \cdots & A_{1,n}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ A_{i,1} & \cdots & A_{i,i} & \cdots & A_{i,n}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ A_{n,1} & \cdots & A_{n,i} & \cdots & A_{n,n}\end{vmatrix}=\begin{vmatrix}A_{1,1} & \cdots & A_{1,i} & \cdots & A_{1,n}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ B_{i,1} & \cdots & B_{i,i} & \cdots & B_{i,n}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ A_{n,1} & \cdots & A_{n,i} & \cdots & A_{n,n}\end{vmatrix}+\begin{vmatrix}A_{1,1} & \cdots & A_{1,i} & \cdots & A_{1,n}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ C_{i,1} & \cdots & C_{i,i} & \cdots & C_{i,n}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ A_{n,1} & \cdots & A_{n,i} & \cdots & A_{n,n}\end{vmatrix}$$

这启发我们将上面每一行都拆成 $-\frac{n-x}{m}\quad\cdots\quad-\frac{n-x}{m}\quad\cdots\quad-\frac{n-x}{m}$ 和 $0\quad\cdots\quad n\quad\cdots\quad 0$

虽然看上去我们把原来的行列式拆成了 2^{m-x} 个行列式相加，但是由于两行相等的行列式值为 0，所以不能选两次及以上第一种，于是就变成了 $m-x$ 个这种

$$\left|\begin{array}{ccccccc}n & 0 & 0 & \cdots & 0 & \cdots & 0\\ 0 & n & 0 & \cdots & 0 & \cdots & 0\\ 0 & 0 & n & \cdots & 0 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\ \hline -\frac{n-x}{m} & -\frac{n-x}{m} & -\frac{n-x}{m} & \cdots & -\frac{n-x}{m} & \cdots & -\frac{n-x}{m}\\ \hline \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & 0 & \cdots & n\end{array}\right|=-\frac{n-x}{m}n^{m-x-1}$$

和一个这种

$$\begin{vmatrix}n & 0 & 0 & \cdots & 0\\ 0 & n & 0 & \cdots & 0\\ 0 & 0 & n & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & n\end{vmatrix}=n^{m-x}$$

所以答案就变成了

$$m^{n-x}\left(n^{m-x}-(m-x)\frac{n-x}{m}n^{m-x-1}\right)=m^{n-x}n^{m-x}-(n-x)(m-x)m^{n-x-1}n^{m-x-1}$$

最终答案就变成了

$$\sum_{x=0}^{\min(n,m)}\binom{n}{x}\binom{m}{x}f_x\left(m^{n-x}n^{m-x}-(n-x)(m-x)m^{n-x-1}n^{m-x-1}\right)$$

单次询问可以 $O(n)$ 算出

Ex

其实前半部分预处理 f 数组也可以移系数做到 $O(n)$，所以这道题就算在每组询问 k 不同时，也可以做到 $O(\sum n)$ 的复杂度，但笔者认为没有必要再考这个了，就没考

$T2$

发现 $d(ij)$ 并不能写成什么和 $d(i),d(j),d(\gcd(i,j))$ 相关的式子，所以一般的套路都是用 $d(ij)=\sum\limits_{x|i}\sum\limits_{y|j}[x\perp y]$ ，但那样笔者并没有想到 100 pts 做法，所以考虑硬靠 $\gcd(i,j)$ 拆式子

考虑 $i=p_1^{\alpha_1}p_2^{\alpha_2}p_3^{\alpha_3}\cdots,p_k^{\alpha_k},j=p_1^{\beta_1}p_2^{\beta_2}p_3^{\beta_3}\dots p_k^{\beta_k}$，有

$gcd(i,j)=p_1^{\min(\alpha_1,\beta_1)}p_2^{\min(\alpha_2,\beta_2)}p_3^{\min(\alpha_3,\beta_3)}\dots p_k^{\min(\alpha_k,\beta_k)}$

同时还有一个式子，若 $n=p_1^{l_1}p_2^{l_2}p_3^{l_3}\dots p_k^{l_k}$，有

$$d(n)=\prod_{x=1}^{k}(l_x+1)$$

这启发了我一个做法，现在我需要约定 $L_p(n)$ 表示 n 中 p 质因子的次数

我们现在枚举了 g，但我们不像正常莫反一样只有 $g|i,g|j$ 的限制，还要加一条

若 $g=p_1^{l_1}p_2^{l_2}p_3^{l_3}\dots p_k^{l_k}$，则应该满足 $\forall x\in\{1,2,3,\cdots,k\},l_x=\min(L_{p_x}(i),L_{p_x}(j))$

发现此时 $L_{p_x}(i)$ 和 $L_{p_x}(j)$ 中至少有一个等于 l_x，我们考虑将等于 l_x 的这个直接消掉，将另一边的贡献加上 l_x

这时候，有一个非常巧妙地做法，是这样的

> 对于 i，我们令
>
> $st=\sum\limits_{x=1}^{k}[L_{p_x}(i)=l_x]2^{x-1},w=\prod\limits_{x=1}^{k}\frac{1}{(L_{p_x}(i)+1)^2}\prod\limits_{L_{p_x}(i)\neq l_x}(L_{p_x}(i)+l_x+1)^2,a_{st}:=a_{st}+w\cdot d^2(i)$
>
> 对于 j，我们令
>
> $st=\sum\limits_{x=1}^{k}[L_{p_x}(j)=l_x]2^{x-1},w=\prod\limits_{x=1}^{k}\frac{1}{(L_{p_x}(j)+1)^2}\prod\limits_{L_{p_x}(j)\neq l_x}(L_{p_x}(j)+l_x+1)^2,b_{st}:=b_{st}+w\cdot d^2(j)$
>
> 通过高维前缀和，使得 $a_x:=\sum\limits_{x\subset y}a_y\prod\limits_{z\notin x\wedge z\in y}(2l_y+1)^2$
>
> $contri=\sum\limits_{x=0}^{2^k-1}a_xb_{x\oplus(2^k-1)}$，其中 $\oplus$ 表示按位异或

发现这个 $contri$ 就是满足这些限制算出来的，但是貌似这些限制并不能使得 $\gcd(i,j)=g$，因为有可能 i,j 还会存在一些 g 没有的共同质因子，所以还要容斥

发现对于 g，只需要减掉所有 2^k-1 个真子集的贡献即可（并非本次算多的，而是前面算多的），于是我们对 g 每一个真子集 g' 按 g 的限制和 g' 的系数算出来答案减掉即可

反正创过去了，注意细节，复杂度大概是数论的 $O(n\lg^2 n)$

$T3$

容易发现四边形不等式，k 段想到 wqs 二分

然后使用决策单调性优化，发现 $w(l,r)$ 是可以通过前缀和达到 $O(1)$ 查询，所以单次复杂度是 $O(n\lg n)$

这个前缀和简单说一下，我们记录前缀 H 和 S 的数量记作 pre_h 和 pre_s

对于 K，定义 $w_0(x) = x - pre_s, w_1 = pre_h(x - pre_s)$ 两个一次函数，求前缀和为 pre_0 和 pre_1

发现 $w(l,r) = pre_{1,r}(pre_{s,r}) - pre_{1,l}(pre_{s,r}) - pre_{h,l}(pre_{0,r}(pre_{s,r}) - pre_{0,l}(pre_{s,r}))$，可能会因为开闭区间等细节有较小出入

整体复杂度便是 $O(n \lg^2 n)$