# 2019 Multi-University Training Contest 10, Editorial

By Quailty and His Friends

## Problem A. Minimum Spanning Trees

Author: Yaohui Zeng

记 $f_{i,j}$ 表示 $i$ 个点的图由权值 $\leq t$ 的边连通的概率，初值 $f_{0,1}=1, f_{0,j}=0\ (2\leq j\leq n)$。

假设 $f_{i,j}\ (0\leq i\leq t-1, 1\leq j\leq n)$ 均已求出，当前考虑到权值为 $t$ 的边。考虑一个 $dfs$ 由 $\leq t$ 的边构成的连通块的过程：从 1 号点所在的由 $\leq t-1$ 的边构成的连通块出发，按照从小到大的顺序递归访问其他由 $\leq t$ 的边构成的连通块（大小相同则优先访问块内最小编号较小的）。现在 $dp$ 这个 $dfs$ 过程，记 $g_{t,i,j}$ 表示从 1 号点所在的由 $\leq t-1$ 的边构成的大小为 $i$ 的连通块出发，已经访问过的其他由 $\leq t$ 的边构成的连通块大小之和是 $j$ 的概率，按照由 $\leq t$ 的边构成的连通块的大小 $s$ 从小到大考虑，先计算出 $f_{t,s}$，再枚举 $g_{t,i,j}$ 挂上多少个大小为 $s$ 的连通块，需要和挂上的连通块之间连 $\geq t$ 的边或者不连边，但是至少有一条权值为 $t$ 的边，并且挂上的连通块之间两两不能连权值 $\leq t$ 的边。不难分析这样 $dp$ 一次的复杂度是 $\mathcal{O}(kn^3\log n)$。

至此还没有考虑最小生成树，为了避免在状态里额外记录权值和，每挂上一个大小为 $s$ 的连通块就给概率额外乘上 $x^{s-1}$，最后会得到一个关于 $x$ 的多项式，其中 $x^t$ 项的系数就是最小生成树是 $t+(n-1)$ 的概率，分别取 $x=0,1,2,3,...,(k-1)(n-1)$ 代入求值，最后使用插值或者高斯消元求出多项式即可。

## Problem B. Line Graphs

Author: Yaohui Zeng

众所周知 $k=0$ 时在 $G$ 中求解最大团是个 NP-Hard 问题，但是 $k=1$ 时只有 $G$ 中的共点边或者三元环才能构成一个 $L(G)$ 中大小 $\geq 2$ 的团，因此在 $L^k(G)$ 求解最大团时只需要考虑 $L^{k-1}(G)$ 中的点（计算其度数）或者三元环。

首先分情况考虑 $L^{k-1}(G)$ 中每个点的度数，

- $k=1$，直接计算；
- $k=2$，$L(G)$ 中每个点对应 $G$ 中一条边 $(u,v)$，度数是 $deg_u+deg_v-2$，也可以直接计算；
- $k=3$，$L^2(G)$ 中每个点对应 $L(G)$ 中一条边，对应 $G$ 中一对共点边 $((u,v),(u,w))$，其中 $v\neq w$，度数是 $2deg_u+deg_v+deg_w-6$，可以枚举 $u$ 来计算；
- $k=4$，$L^3(G)$ 中每个点对应 $L^2(G)$ 中一条边，对应 $L(G)$ 中一对共点边，对应 $G$ 中一对有公共边的共点边对 $(((u,v),(u,w)),((u,v),(v,x)))$，其中 $u\neq x, v\neq w$，度数是

$3deg_u + 3deg_v + deg_w + deg_x - 14$，或者是 $(((u,v),(u,w)),((u,v),(u,x)))$，其中 $v,w,x$ 两两不同，度数是 $4deg_u + 2deg_v + deg_w + deg_x - 14$，两种情况均可以枚举 $(u,v)$ 来计算。

预处理每个点的相邻点中最大的三种点度及其出现次数，即可在关于 $n$ 和 $m$ 线性的时间内完成这一部分计算。

但是如果此时计算出的最大团大小 $\leq 3$，还需要考虑 $L^{k-1}(G)$ 中的三元环，不难证明此时 $G$ 中每个点的度数都不超过 3，除非 $k=2$，此时允许某些连通块是完全二分图 $K_{1,4}$，进一步可以证明迭代一次至多使边数乘以 2，于是可以直接模拟出 $L^{k-1}$ 之后枚举所有三元环，这部分计算也是在关于 $n$ 和 $m$ 线性的时间内完成的。标程采用的方法是当 $k \geq 2$ 时模拟出 $L^{k-2}(G)$ 之后枚举三元环并计算三条边共点的方案数，实际上没有差别。

最后需要注意的是，最大团大小为 0 时方案数是 1（已经包括在样例中），大小为 1 时上述方法会计算每个团两次，需要将答案除以 2。

## Problem C. Valentine's Day

Author: Jingbang Chen

记已经购买的礼物一次都不能让女朋友开心的概率为 $s_0$，恰好开心一次的概率为 $s_1$，再买一个概率为 $p$ 的礼物会使 $(s_0,s_1) \to (s_0p, s_1 + (s_0 - s_1)p)$。

显然，当 $s_0 \leq s_1$ 时加入任何礼物都不会使答案变优。类似地，如果删除一个礼物之后 $s_0 \leq s_1$，那么删除这个礼物不会使答案变得更差。因此，任意一个最优解对应的非空礼物集合 $G$ 都满足 $s_0 \leq s_1$，且该集合每个真子集均有 $s_0 > s_1$，也即 $\forall_{T \subsetneqq G}, \frac{s_1}{s_0} = \sum_{p \in T} \frac{p}{1-p} < 1$。

不难发现 $\frac{p}{1-p}$ 和 $p$ 的单调性相同，因此，如果一个满足上述条件的非空礼物集合 $G$ 不是由使女朋友开心概率最大的若干个礼物构成，那么可以把集合内概率最小的礼物 $a$ 换成集合外概率最大的礼物 $b$，从而得到一个答案不会更劣的礼物集合 $(G - \{a\}) \cup \{b\}$。设新集合中概率最小的礼物为 $c$，新集合的真子集 $((G - \{a\}) \cup \{b\}) - \{c\}$ 可能不满足 $s_0 > s_1$，此时可以不断地删去集合中概率最小的礼物，直到该集合满足条件。这个过程不会使答案变劣，不断地替换和删除礼物可以得到最优解。

最终的结论是，按照概率从大到小的顺序购买礼物直到 $s_0 \leq s_1$ 或者全部买完为止，即可得到一个最优解。

## Problem D. Play Games with Rounddog

Author: Dongyang Wang

Nim 游戏先手必胜条件是石子的异或和非零，如果 Calabash 想赢，必须使 RoundDog 无论怎么选石头，异或和都非零，这表明 Calabash 对于 $SA$ 的选取会使得对应的 $W$ 构成一组线性基。由于线性基是拟阵，要求线性基内的元素总和最大，可以按照 $W$ 从大到小的顺序加入线性基。

题目中限制了 Calabash 选出来的字符串必须是以 $T$ 为后缀，这些字符串在后缀自动机上的节点均位于 parent 树上 $T$ 所在节点的的子树内，于是可以先建出原串的后缀自动机，再按照 $W$ 从大到小的顺序，从对应的点开始沿着 parent 指针不断往上跳，沿途向每个点的线性基添加该元素，直到某个点的线性基加不进该元素或者根节点的线性基也加入了该元素为止，这部分复杂度是 $\mathcal{O}(n \log^2 W)$。

需要注意，$W$ 的范围设置成 $2^{58}$ 会使得最终结果超出有符号 64 位整数的表示范围，但仍在无符号 64 位整数的表示范围内。

# Problem E. Welcome Party

Author: Shaoyuan Chen

题意即为给定 $n$ 个数对 $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$，将 $\{1, 2, \cdots, n\}$ 划分为两个不相交的非空子集 $S$ 和 $T$，使得 $|\max_{i\in S} x_i - \max_{j\in T} y_j|$ 最小。

称 $\hat{i} = \arg\max_{i\in S} x_i$ 为 $x$ 临界的，对 $y$ 临界也类似定义。枚举 $x$ 临界的下标 $i$，那么 $j$ 可能成为 $y$ 临界的，当且仅当 $y_j \geq \max\{y_k : k \neq i, j, x_k > x_i\}$。将所有数对以 $x$ 为关键字排序后，就可以用数据结构维护 $y$ 并找出最接近 $x_i$ 的 $y_j$，且 $y_j \geq \max\{y_k : k \neq i, j, x_k > x_i\}$，复杂度是 $\mathcal{O}(n \log n)$。

# Problem F. Dense Subgraph

Author: Ildar Gainullin

首先观察到，从所有直径 $\leq 2$ 的子图中一定可以找到 density 最大的子图，否则沿着直径中间的边断开可以得到两棵至少有两个点的树，其中至少有一棵树的 density 不会更小。

于是只需要考虑所有直径 $\leq 2$ 的子图，只有 $\mathcal{O}(n2^{deg})$ 个，每一个满足 density $> x$ 子图中所有点不能同时亮。对每个点记录父亲、自己、以及每个儿子的状态，预处理出所有不合法状态之后在树上 $dp$ 即可。

# Problem G. Closest Pair of Segments

Author: Yaohui Zeng

二分答案之后将线段扩张成“香肠”，检查是否有一对“香肠”相交。

考虑对 $x$ 扫描线，用 $set$ 维护这些香肠关于 $y$ 的顺序，那么总是只需要检查相邻的两个“香肠”是否相交。亦即，对于一个插入事件，检查新的“香肠”是否和上下相邻的两个“香肠”相交，对于一个删除事件，删除“香肠”之后检查上下两个相邻的“香肠”是否相交。如果判定到相交说明答案偏大，否则答案偏小。

由于在判定到相交之前的扫描过程中，“香肠”都是互不相交的凸图形，可以简单地用一条连接“香肠”最左点和最右点的线段来描述“香肠”在扫描线上的相对顺序，判定“香肠”相交时直接计算扩张前的两个线段之间的距离即可。

复杂度是 $\mathcal{O}(n \log n \log(1/\epsilon))$。

# Problem H. Coins

Author: Hao Ling

先将硬币组分成两类，第一类 $a_i > b_i$，第二类 $a_i \leq b_i$，并记 $g(x)$ 为从第一类硬币中符合限制地取出 $x$ 枚硬币的最大价值和，令 $h(x)$ 为从第二类硬币中符合限制地取出 $x$ 枚硬币的最大价值和。

要求出 $g$，只需要将第一类的所有硬币按价值从大到小排序，则 $g(x)$ 为前 $x$ 个硬币的和。因为 $a_i > b_i$，可以发现在这个贪心方法中，如果选了 $b_i$，则一定会先选了 $a_i$；

要求出 $h$，需要观察到：第二类硬币中最多只有一组硬币，会选了 $a_i$ 而没有选 $b_i$。因为如果有两组硬币 $i, j$, 满足 $a_i \le b_i, a_j \le b_j$，且某个方案只选了 $a_i, a_j$ 而没有选 $b_i, b_j$，不失一般性，令 $a_i \ge a_j$，那么有 $b_i \ge a_i \ge a_j$，所以选 $a_i, b_i$ 而不选 $a_j, b_j$ 是个更优解，这时需要按 $a+b$ 的值从大到小将第二类的每组硬币排序，则 $h(2x)$ 就是前 $x$ 组硬币的和，$h(2x+1)$ 就是"前 $x$ 组加上不在前 $x$ 组里的最大的 $a$"以及"前 $x+1$ 组中减去前 $x+1$ 组中最小的 $b$"两种方案中的较大值。

求出 $g, h$ 之后, 立即有 $f(x) = \max_{x_1+x_2=x} g(x_1)+h(x_2)$，朴素地计算这个 $max$ 卷积只能做到 $\mathcal{O}(n^2)$，但是观察到 $g$ 是个单调递增的凸函数而 $h$ 是个单调递增函数，设 $p(x) = \min(\arg\max_{x_0} g(x-x_0)+h(x_0))$，可以证明 $p(x)$ 有单调性，利用决策单调性即可优化到 $\mathcal{O}(n \log n)$。

# Problem I. Block Breaker

Author: Minghong Gao

每个砖块只会直接影响其周围 4 个格子，每当有一个砖块被敲掉时就检查周围的 4 个格子看看是否会对其造成影响，敲掉之后再递归检查下去。由于每个砖块只会被敲掉一次，复杂度是 $\mathcal{O}(nm+q)$。

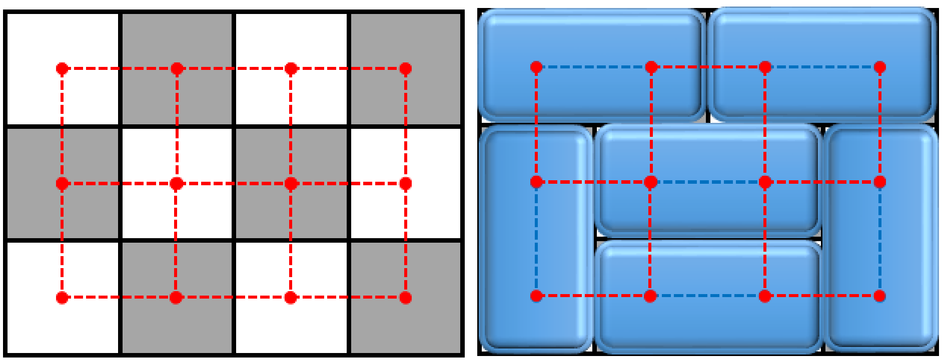
# Problem J. Domino Covering

Author: Jingzhe Tang

当 $n=2$ 时，答案是斐波那契数列的第 $(m+1)$ 项；类似地可以发现，对于固定的 $n$，答案是某个齐次线性递推数列的第 $m$ 项。很可惜，这个递推数列的每一项会和前面至多 $2^{\lceil n/2 \rceil}$ 项有关，而当项数很大时，很难设计算法通过此题。

废话不多说，开始讲解这道题的解题思路，为了解决这道题，你需要有一定程度的图论和线性代数基础。

我们可以将网格黑白染色，使得黑色格子的邻居都是白色格子，反之亦然。我们可以把格子作为某个无向图中的点，在相邻的格子对应的点之间连一条无向边，从而把问题规约到统计二分图的完美匹配方案数上。



然而，一般的二分图是比较难计算完美匹配方案数的，所幸网格图还是平面图，而平面图的某些性质使得我们可以在关于点数的多项式时间内计算出方案数。

这里需要引入矩阵的 Pfaffian 值的定义。一个矩阵 $A$ 是斜对称的，当且仅当对于任意的 $A_{i,j}$ 有 $A_{i,j} = -A_{j,i}$。任意一个矩阵的行列式都可以写成一个关于矩阵元素的多项式，而一个斜对称矩阵的行列式总能写成一个多项式的平方，这里的这个多项式的值（也即 $\pm\sqrt{\det(A)}$，这个矩阵的行列式的某个平方根）被称为这个矩阵的 Pfaffian 值（简称 Pf），例如

$$\mathrm{Pf}\begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix} = 0, \mathrm{Pf}\begin{bmatrix} 0 & a & b & c \\ -a & 0 & d & e \\ -b & -d & 0 & f \\ -c & -e & -f & 0 \end{bmatrix} = af - be + cd.$$

正式的定义如下：对于 $n \times n$ 的斜对称矩阵 $A$，若 $n$ 为奇数，则 $\mathrm{Pf}(A) = 0$，否则

$$\mathrm{Pf}(A) = \sum_{M \in PM(n)} \mathrm{sgn}(M) \prod_{(i,j) \in M} A_{i,j},$$

这里 $PM(n)$ 表示将 $\{1, 2, \ldots, n\}$ 划分为 $\frac{n}{2}$ 个不相交的无序二元组的所有方案组成的集合，而 $\mathrm{sgn}(M)$ 在 $M = \{(i_1, j_1), (i_2, j_2), \ldots, (i_{n/2}, j_{n/2})\}$ ($i_1 < i_2 < \ldots < i_{n/2}$, $i_1 < j_1$, $i_2 < j_2$, …, $i_{n/2} < j_{n/2}$) 时表示排列 $(i_1, j_1, i_2, j_2, \ldots, i_{n/2}, j_{n/2})$ 的逆序对数。

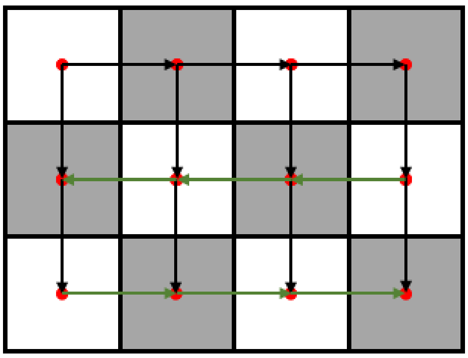
不难注意到，如果去掉上式中的 $\mathrm{sgn}(M)$，我们可以得到平面图完美匹配的方案数

$$\mathrm{PerfectMatching(G)} = \sum_{M \in PM(n)} \prod_{(i,j) \in M} E_{i,j},$$

这里 $G$ 为一个 $n$ 点无向平面图，$E$ 为 $G$ 的邻接矩阵（$E_{i,j}$ 表示点 $i$ 到点 $j$ 的无向边数）。而任意一个无向平面图 $G$ 都存在一种给边定向（无向边变成有向边，入边视为 $-1$，出边视为 $1$）的方案，使得定向后的邻接矩阵 $E'$ 满足 $\mathrm{PerfectMatching(G)} = |\mathrm{Pf}(E')|$，这样我们就可以把计算完美匹配的方案数规约到计算定向邻接矩阵行列式的平方根上（当然，也需要确定绝对值的符号）。

不难证明这样的 $E'$ 一定存在。任意定向都能使得邻接矩阵变成斜对称矩阵（），而我们只需要使得平面图的任意一个环上包含奇数条顺时针边就可以将其变成 $E'$。这等价于使得包围每个面的环上包含奇数条顺时针边，于是我们可以首先将 $G$ 的一棵生成树上的所有边任意定向，然后在 $G$ 的平面对偶图上删去这些边，剩下的边将关于对偶图上的点（也即原图上的每个面）形成一棵树，此时只需要不断地将这个树的叶节点（不含根节点）对应的面上唯一一条没有定向的边按照满足要求的方式定向，再删除这个叶节点即可。

这个定向的过程比较简单，例如我们可以将 $n \times m$ 网格图做如下定向使得 $\mathrm{PerfectMatching(G)} = \mathrm{Pf}(E')$:（以下均以 $n = 3, m = 4$ 为例）



不妨将第 $i$ 行的第 $j$ 个点编号为 $(i + (j-1)n)$，则相应的定向邻接矩阵如下所示：

$$E' = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} U & V & \mathbf{0} & \mathbf{0} \\ -V & U & V & \mathbf{0} \\ \mathbf{0} & -V & U & V \\ \mathbf{0} & \mathbf{0} & -V & U \end{bmatrix},$$

这里 $U$ 和 $V$ 是两个 $n \times n$ 矩阵。

利用其分型的性质，我们可以归纳证明该矩阵的行列式等于 $n \times n$ 的矩阵 $F_m$ 的行列式乘以 $(-1)^{nm/2}$，这里 $F_0 = I_{n\times n}, F_1 = U, F_k = U \cdot F_{k-1} - F_{k-2}\ (k \ge 2)$。例如当 $n = 3$ 时，

$$F_4 = \begin{bmatrix} 6 & 0 & -5 \\ 0 & 11 & 0 \\ -5 & 0 & 6 \end{bmatrix}, \det(F_4) = 11^2.$$

由于 $U$ 的特殊性，我们不难证明 $F_m$ 内只有一半的元素可能非零，而且这些元素的下标 $(i, j)$ 满足 $(i + j)$ 的奇偶性相同，这意味着我们在使用消元法计算 $F_m$ 的行列式时，奇数行和偶数行互不影响。通过进一步的观察可以发现，只考虑奇数行或偶数行的非零元素时，分别算出的行列式取绝对值后是相等的，这意味着我们只需要计算一半的行列式（以及它的正负号）就可以得到行列式绝对值的平方根。

但是利用快速幂直接计算 $F_m$ 是 $\mathcal{O}(n^3 \log m)$ 的，无法通过此题。注意到 $F_m$ 可以表示成一个 $2n\times 2n$ 矩阵 $P$ 的 $m$ 次幂的某个子矩阵，这意味着序列 $F_0, F_1, \ldots$ 存在项数不超过 $2n$ 的齐次线性递推，因此我们可以计算出矩阵 $P$ 的化零多项式，将 $F_m$ 表示出 $F_0, F_1, \ldots, F_{2n-1}$ 的加权表示。由于 $U$ 只有 $\mathcal{O}(n)$ 个非零元素，所以计算 $F_0, F_1, \ldots, F_{2n-1}$ 可以在 $\mathcal{O}(n^3)$ 的时间内完成，我们只需要在 $\mathcal{O}(n^2)$ 的时间内计算出矩阵 $P$ 的化零多项式，再利用快速幂在 $\mathcal{O}(n^2 \log m)$ 的时间内计算出它们的加权系数即可。

具体来说，我们可以令

$$P = \begin{bmatrix} U & -I_{n\times n} \\ I_{n\times n} & \mathbf{0} \end{bmatrix},$$

则有

$$P^m = \begin{bmatrix} F_m & -F_{m-1} \\ F_{m-1} & F_{m-2} \end{bmatrix},$$

此处 $F_{-1} = \mathbf{0}$。

而 $P$ 的化零多项式 $f_{2n}(\lambda) = \det(\lambda I_{2n\times 2n} - P) = \lambda^{2n} + \sum_{i=0}^{2n-1} a_i\lambda^i$ 所涉及到的行列式也具有分型的性质，不难归纳得出 $f_{2n}(\lambda)$ 可以从 $f_{2n-2}(\lambda)$ 和 $f_{2n-4}(\lambda)$ 计算得到。

计算出化零多项式 $f_{2n}(\lambda)$ 后，我们便可以知道 $F_k + \sum_{i=0}^{2n-1} a_i F_{k-2n+i} = \mathbf{0}$ 对于任意整数 $k \geq 2n$ 成立，所以我们可以通过计算 $\lambda^m \bmod f_{2n}(\lambda)$ 得到 $F_m$ 关于 $F_0, F_1, \ldots, F_{2n-1}$ 的加权表示。

顺带一提，你也可以尝试利用 Temperley & Fisher 和 Kasteleyn 分别在 1961 年得到的如下表达式来计算答案。

$$\prod_{j=1}^{\lceil m/2\rceil}\prod_{k=1}^{\lceil n/2\rceil}\left(4\cos^2\left(\frac{\pi j}{m+1}\right)+4\cos^2\left(\frac{\pi k}{n+1}\right)\right)$$

不过你需要利用第二类 Chebyshev 多项式将其转化到计算一个 $n \times n$ 的矩阵的行列式的平方根。本质上来说，这个做法和前述做法没有差异，你可以通过阅读我们提供的标程来了解这种做法。

总结一下本题解所说的做法。

- 首先构造 Pfaffian 定向，找出 $F_m$ 的递推式，递推计算出模意义下的 $F_0, F_1, \ldots, F_{2n-1}$；
- 然后利用分形技巧找出 $P$ 的化零多项式 $f_{2n}(\lambda)$，快速幂计算模意义下的多项式 $(\lambda^m \bmod f_{2n}(\lambda))$；
- 最后加权得到 $F_m$，消元法计算模意义下的 $\sqrt{(-1)^{nm/2}\det(F_m)}$。

标程的整体时间复杂度为 $\mathcal{O}(n^3 + n^2\log m + n\log p)$。

最后是题外话，这道题存在一个有趣的现象：令 $\text{ways}(n+1, m+1)$ 表示 $n \times m$ 网格图的答案，则当 $\text{ways}(p,q) \neq 0$, $\text{ways}(u,v) \neq 0$, $p|u$, $q|v$ 时有 $\text{ways}(p,q)|\text{ways}(u,v)$。欢迎读者提供证明。

# Problem K. Make Rounddog Happy

Author: Dongyang Wang

考虑分治，每次统计跨过中点的合法区间个数。

枚举最大值所在一侧的区间端点，可以计算出另一个端点的合法区间，需要满足三个限制：

- 不能出现更大的数字，使用单调栈预处理每个数字两侧最近的不小于它的数的位置；
- 不能有重复数字，在前边预处理的基础上额外限制不出现重复数字；
- 区间长度要足够长，由最大值决定。

复杂度是 $\mathcal{O}(n\log n)$。