四个题都偏向思维难度，代码难度均较低。

A

显然差分一下，问题变成每次可以反转某两个位置，问最后能不能把所有位置变成 0.

建图，如果可以翻转两个位置则在它们之间连一条边。显然每个连通块内 1 的个输都必须是偶数才有解，连通块之间独立。

对于每个连通块，随便找一棵生成树，我们会发现仅翻转这棵树上的边就一定可以达到目标状态（dfs 之后从叶结点往上确定每条边是否被选择），并且方案是唯一的。于是我们做到了构造一组方案。

由于取出一棵生成树后方案唯一，因此非树边可以任选。方案数为 $2^{\text{非树边数量}}$。

B

考虑把原题意变为：每次可以选择 (i,j)，让第 i 行和第 j 列上的数字同时加上任意的数字 c，问能否把矩阵变成 0。

$n=1$ 或 $m=1$ 时该问题是平凡的，所以不予讨论。其他情况都永远有解。

显然可以先在 $(1,1)$ 上做该操作，把矩阵的总和变成 0。接下来我们的目标是把每行每列的总和都变成 0。

不妨设当前的矩阵为 A，我们现在的目标是把每行的和都变成 0，在 $(i,1)$ 位置做 $c=-\frac{1}{m-1}\sum_{j=1}^{m}A_{i,j}$ 的操作，我们会发现这 n 次操作执行完之后，第一列完全不会变化（c 的总和为 0），后面每一列的总和也完全不会发生变化，并且每一行的总和都恰好变成了 0（后面 $m-1$ 列每一列都减掉了类似于平均数的东西）。

于是再对列做这样的操作，每一列的总和也会变成 0。

于是对于一个每行每列和都是 0 的矩阵，我们可以对于所有位置 $(i,j),i,j>1$，让 (i,j) 和 $(1,1)$ 操作 $A_{i,j}$，$(1,j)$ 和 $(i,1)$ 操作 $-A_{i,j}$ 会让 (i,j) 变成 0，并且每行每列的总和不变。于是从右下到左上把除了第一行第一列的数字都变成 0，此时由于每行每列总和都是 0，第一行第一列的数字也自然都全是 0。

由于这 nm 种线性操作可以到达所有矩阵的状态（所有矩阵构成的是 nm 维的线性空间），因此答案事实上是唯一的。

C

显然行列独立，于是对于每行/每列 hash 之后变成了一维（$n=1$）的问题。

下面编号都指的是空隙的编号，最左侧的边界编号为 0。

定义：一个回文中心 x（回文串半径 l）“左可达”，当且仅当存在 $[x-l,x)$ 的左可达回文中心。其中 0 位置是左可达的。

同理可定义右可达。

定理：若 i 左可达，j 右可达且 $i<j$，则存在一种方式可以把回文串翻折成区间 $[i,j]$（即 i,j 之间的字母构成的回文串）。

读者自证不难。提示：证明每次都翻一个离边界最近的回文中心一定合法。

于是跑两次 manacher，前缀和统计一下答案即可。

D

正常做法：

注意到 NO 当且仅当，每个点的所有邻居的导出子图是若干个团。压位即可做到 $O(\frac{nm}{w})$（每次加入一个邻居，若它不在当前的团里就新建一个团，否则判断一下它是不是和某个团里所有点都有边且和其它团完全没有边）。

Bonus

该问题事实上和矩阵乘法复杂度等价，即可以做到 $O(n^{\omega})$，其中 $\omega\approx 2.32$。

考虑记邻接矩阵为 A，那么 A^2 表示一个点走恰好两步走到另一个点的方案数。考虑下面的式子：

$$\sum_{(u,v)\in E}\binom{A_{u,v}^2}{2}$$

注意到 K_4 会在上式中被统计 6 次，而一个“钻石”会被恰好统计一次。因此 $\bmod 6$ 之后判断余数是否为 0 就**有一定概率**可以判断出是否存在钻石。

于是多随机几次，每次随机删掉若干个点运行上述算法即可。