矩阵 matrix：

设初始状态含有白格子的列的数量为 c，设 k 表示最少的操作次数使得某一行的所有格子变成黑色，那么在 k 次操作之前，每一行都是不全为黑的，所以每次操作不会减少 c，而且为了步数最少，c 肯定也不会增加，因此我们只要找到 k，那么答案就是 k+c。

考虑枚举全黑的那一行 r，设 x 表示这一行白格子数量：

如果列 r 有黑格子，那么最少次数为 x，只要把列 r 有黑格子的那一行对每个白格子所在列进行操作即可。

如果没有，我们需要用一次操作来让这一列含有黑格子，所以要 x+1 步。

时间复杂度 O(n^2)

猜数列 hidden：

只考虑向右的话，用 f[S][i][j]表示当前已经处理了 S 中的条件，现在正在处理第 i 个条件且完成了前 j 个数，还需要多少个数。转移分两种情况：（1）填入第 i 个条件的下一个数，从 f[S][i][j+1]转移；（2）枚举条件 k 使得只要接上 k 就一定能满足 i，从 f[S|(1<<i)][k][0]转移。

对于向左只要类似地加上两维[k][l]表示左侧正在处理第 k 个条件，完成了第 l~x[k]个数。转移时增加三种情况：（3）填入第 k 个条件的下一个数，从 f[S][i][j][k][l-1]转移；（4）若第 i 个条件和第 k 个条件的下一个数相同，从 f[S][i][j+1][k][l-1]转移；（5）若 l=1，即第 k 个条件已经填完，枚举 u，v 使得第 u 个条件完成 v 个数以后接上 k 就能满足，从 f[S|(1<<k)][i][j][u][v+1]转移。

特殊情况：i=0 表示右侧尚未开始，k=0 表示左侧已经结束。

注意填数时要保证不会违反另一个条件，即(1)填的数必须是条件 k 的前 l 个数，(3)填的数必须是条件 i 的前 j 个数。

由于转移顺序并不明显，需要用记忆化搜索。

时间复杂度 O(2^n*n^3*10^2)

围墙 c：

我们将置换循环分解，然后所有大小为 2 的循环可以直接确定，剩下的写个 dfs 搜即可。

时间复杂度 O(2^(n/4))