泉轻流花暖（yumemi）

> std 长度：6123 Bytes

首先把前缀和调对，即要求所有前缀和非负。维护一个堆，如果某个时刻 $s_i < 0$，我们就找到堆里面最小的 a_j，且满足 s_j 是左括号，然后把他改过来；接下来把总和调对，这时需要把 `( -> )`，这个相当于 i 后面不能超过 $s_i/2$ 个 `( -> )`。从小到大依次考虑每个 a_i，如果能填就填，一定是最优的。

我们发现两个过程几乎是独立的：找到全局 s 的最小值 p，那么第一个过程一定全部发生在 p 之前，第二个过程一定全部发生在 p 之后（因为只要能把 s_p 修改到 $s_p \geq 0$ 显然后面的也 ≥ 0 了；最后的 s 中一定有 $s_p = 0$，所以第二个过程的修改只能在 $> p$ 的位置进行）。

考虑修改一个 a_i，有什么影响。首先我们考虑第一个过程，发现 s_i 是不会变的，只是修改了哪些 a_i 可能会改变。我们按照 i 去找 j 可能不太好找，考虑提前预处理出来这样走一遍会有 $s_i < 0$ 的 i，称这些点为匹配点，那么相当于每个匹配点 i 会往前匹配一个 j，求权值和最小的完美匹配。

发现此时问题就变成了人员调度，不过他是在链上做，所以 1log 也很容易，讨论一下最优匹配集合中至多一个点的变化即可。总复杂度 $O((n+q)\log n)$。

岁云暮矣待月归（kofuru）

> std 长度：6367 Bytes

首先考虑 $k=0$ 怎么做。

不难得到一个暴力做法：暴力容斥两条对角线、每行、每列，现在相当于钦定若干位置不能放棋子，假设剩下 c 个位置，方案数就是 $\binom{c}{m}$。总复杂度 $O(4^n n^2)$。然后可能只枚举行可以做到 $O(2^n)$ 左右。

我们考虑：如果没有对角线的限制，那么实际上如果选了 x 行 y 列不能放棋子，被覆盖的格子个数实际上就是 $nx+ny-xy$。

如果只有一条对角线选了呢？这个时候被覆盖的格子个数可以这样算：行列覆盖的格子个数 $nx+ny-xy$，加上对角线覆盖的格子个数 n，再减掉被行列和对角线都盖住的格子个数。第三个我们可以 DP 一下，比如如果钦定的是主对角线，那么就在 DP 的时候每次同时加入第 i 行第 i 列，记录一个 $f(i,j,k,l)$ 表示目前决策到 (i,j)，选了 j 行 k 列，目前行列和对角线都盖住的格子个数为 l 的方案数。最后答案就是

$$\sum_{j,k,l}(-1)^{j+k+1}\binom{n^2-(nj+nj-jk+n-l)}{m}\times f(n,j,k,l)$$

对一个 m 算答案是 $O(n^3)$ 的，这样就得到一个 $O(n^5)$ 的做法。实际上注意到剩下的格子数不会超过 n^2，所以我们对 $s=0\cdots n^2$ 算出剩下 s 个格子的容斥系数和，最后 $O(n^4)$ 求出所有答案即可。

类似地，如果两条对角线都被 ban 了，我们就把第 $i,n-i+1$ 行和 $i,n-i+1$ 两列放到一起考虑，每次考虑这两行两列是否要选中，同样是记录一个 $f(i,j,k,l)$ 的状态。也可以做到 $O(n^4)$。

再考虑 k 个特殊格子怎么办，发现只需要状压一下，多记录一个 S 表示目前选中的行、列、对角线，已经覆盖了 S 集合中的特殊格子。最后算没有被覆盖的格子个数时，要额外减掉 $k-|S|$。这样可以得到一个 $O(2^k n^4)$ 的做法。大概能拿到 80 分。

考虑进一步优化，我们可以先 $O(n^4)$ 对不存在特殊格子的行列 DP 一遍，然后再 $O(2^k k^4)$ 做一遍那些带有特殊格子的行列，最后 $O(n^3k^3)$ 合并两个 DP 状态。

这样总复杂度就是 $O(2^k k^4+n^4+n^3k^3)$。可以获得满分。

慕念萦心间（kemuru）

> std 长度：2879 Bytes

- Subtask 1 的做法

注意到可以把 $AB = BA$ 看作关于 B 的 n^2 个元素的 n^2 条方程，高斯消元求解即可，如果最后自由元的个数是 k，那么答案就是 p^k。（由此也可以看出答案一定是 p 的幂）

时间复杂度 $O(n^6)$，期望得分 20。

- Subtask 2,3 的做法

如果 A 是对角矩阵，发现此时约束简化为 $\forall i, j$，有 $A_{i,i}B_{i,j} = A_{j,j}B_{i,j}$，那么如果 $A_{i,i} = A_{j,j}$ 则 $B_{i,j}$ 可以任选；否则必须有 $B_{i,j} = 0$。由此可以简单计算自由元个数。

到此可以获得 30 分。

- Subtask 4：$A^n = 0$ 的解法

记 $F = \mathbb{F}_p$，将 A 视同 $V = F^n$ 上的线性映射。$\ker(T)$ 定义为所有满足 $Tv = 0$ 的向量 v 的集合。

我们知道有如下的一系列子空间：$\ker(A) \subseteq \ker(A^2) \subseteq \cdots \subseteq \ker(A^n) = V$。

取 $\ker(A)$ 的一组基 $v_{1,1}, \cdots, v_{1,s_1}$，接下来扩充为 $\ker(A^2)$ 的基 $v_{1,1}, \cdots, v_{1,s_1}, v_{2,1}, \cdots, v_{2,s_2}$，依次类推，最后得到 K 的一组基 $v_{i,j}$，其中 $1 \le i \le n, 1 \le j \le s_i$，$s_i = \dim\ker(A^i) - \dim\ker(A^{i-1})$。

事实上，我们可以把基取的「更好看」一点：有 $v_{i,j} \in \ker(A^i), v_{i,j} \notin \ker(A^{i-1})$，且有 $Av_{i,j} \in \ker(A^{i-1})$。于是可以将 A 看作 $\ker(A^i)/\ker(A^{i-1}) \to \ker(A^{i-1})/\ker(A^{i-2})$ 的线性映射（其中 $i \ge 2, A^0 = I$），进一步，这个线性映射还是一个单射。

于是我们得到一些推论：必有 $s_{i-1} \ge s_i$，且我们可以取 $v_{i,1}, \cdots, v_{i,s_i}$ 使得 $Av_{i,j} = v_{i-1,j}$ 对 $1 \le j \le s_i$ 成立。

B 由它在所有 $v_{i,j}$ 上的作用唯一确定。那么 B 只需要满足对任意 i, j，有 $ABv_{i,j} = BAv_{i,j} = Bv_{i-1,j}$。对于 $i \ge 2$ 的情况，一旦确定 $Bv_{i,j}$ 是什么，那么 $1 \le j \le s_i$ 的 $Bv_{i-1,j}$ 就唯一确定了，而对于 $s_i < j \le s_{i-1}$ 的 $Bv_{i-1,j}$ 则可以任意选取；而在 $i = 1$ 的情况，这相当于 $ABv_{1,j} = BAv_{1,j} = 0$，即 $Bv_{1,j} \in \ker(A)$。

进而有 $Bv_{1,j} = ABv_{2,j} \in \ker(A)$，那么必须要有 $Bv_{2,j} \in \ker(A^2)$。综上我们发现，B 可以按照如下的步骤确定：对每个 $1 \le i \le n$，以及 $s_{i+1} < j \le s_i$，我们确定一个 $Bv_{i,j} \in \ker(A^i)$。那么其自由度是多少呢？就是

$$\sum_{i=1}^{n}(s_i - s_{i+1}) \times \sum_{j \le i} s_j = \sum_{i=1}^{n} s_i^2$$

于是，我们只需要求出 $s_i = \dim\ker(A^i) - \dim\ker(A^{i-1})$，答案就是 $\sum_{i=1}^{n} s_i^2$。

这里数据范围只开到 100，允许你做 n 次高斯消元暴力求解所有 $\dim\ker(A^i)$。

- Subtask 5 的解法

这时我们已经给出了一个 V 的直和分解 $V = \oplus_{i=1}^{e_i} \ker((A - \lambda_i I)^{e_i})$

容易证明 A, B 交换时每个 $\ker((A - \lambda I)^{e_i})$ 也必须是 B-不变子空间，因此在每个直和项内部用 subtask 4 的解法即可。具体地在这个直和项内可以把 A 视为 $A - \lambda I$，那么它就幂零了。

至此可以获得 64 分，所用的算法实际上只有矩阵乘法和高斯消元（误）

- 满分解法

如果仅仅追求理论上的解法，我们总可以将 $\mathbb{F}_p$ 进行扩域（扩域自然不改变解空间维数），使得最小多项式分裂后做上三角化，就转化为 subtask 5；然而扩域的代价实在太大了。

首先我们考虑如何求解 A 的有理标准型，我们设 A 的最小多项式是 $m_A(x)=\prod p_i^{e_i}$，可以发现这样一个事实：随机选取 $v\in V$，满足 v 相对与 A 的最小多项式 $m_v(x)$ 大概率就是 $m_A(x)$，实际上这个概率至少是 $1-\frac{n}{p}$。这是因为首先必有 $m_v(x)\mid m_A(x)$，如果二者不相等那么必须存在 p_i 使得 $m_v(x)\mid m_A(x)/p_i$，也就是说 $v\in\ker(m_A(x)/p_i)$，那么 v 的取值至多是 n 个 $n-1$ 维子空间的并，于是取到这其中的概率就 $\le\frac{n}{p}$。

那么我们每次随机选取一个向量 v，注意向量乘矩阵是 $O(n^2)$ 的，我们不断求 $v,Av,A^2v,\cdots$，直到有一个 $A^kv\in\operatorname{span}(v,Av,\cdots,A^{k-1}v)$ 为止；此外到这里我们还可以顺便求出 A 的最小多项式。

接下来在商空间 $V/F[A]v$ 中接着随机选取向量做下去即可。我们容易在 $O(\deg m_v(x)\times n^2)$ 的时间复杂度内完成一次分解，那么总的时间复杂度是不超过 $O(n^3)$ 的。

这个过程至多进行 n 次，那么我们就有 $(1-\frac{n}{p})^n\ge 1-\frac{n^2}{p}$（伯努利不等式）的概率得到 V 的一个循环分解：$V=F[A]v_1\oplus F[A]v_2\oplus\cdots\oplus F[A]v_s$，且我们还顺便求出了所有 v_i 的最小多项式。

这实际上就给出了 A 的不变因子组 $d_k(x)\mid d_{k-1}(x)\mid\cdots\mid d_1(x)$。考虑如何据此求出 s_i。

先假装扩域，使所有 $d_i(x)$ 分裂，我们还是只需要对每个特征值 λ 对应的广义特征子空间去考虑。假设 $d_i(x)$ 中 $(x-\lambda)$ 的系数为 c_i，即 $(x-\lambda)^{c_i}\|d_i(x)$，那么可以发现限制在 $\ker((A-\lambda I)^n)$ 内时，有

$$s_i=\dim\ker(A^i)-\dim\ker(A^{i-1})=\sum_{j=1}^{k}[c_j\ge i]$$

那么有（注意必有 $c_i\ge c_{i+1}$）

$$\sum s_i^2=\sum_i\sum_{x,y}[c_x\ge i][c_y\ge i]=\sum_{x,y}\min(c_x,c_y)=\sum_{j=1}^{k}(2j-1)c_j$$

这就是 λ 对应的解空间维数。

这时候我们发现，并不需要真的求出分裂后的所有多项式，因为上面这个式子是满足可加性的，所以答案其实就是 $\sum_{j=1}^{k}(2j-1)\deg d_j(x)$。综上我们就解决了本题。