T1

假设一个点的两条出边为 i,j，我们新建一个图给 i,j 连边。如果一个点的两条入边为 i,j，我们也给 i,j 连边。

不难发现新图上每个点度数恰好为二，并且只有偶环。我们的要求事实上就是在新图上选 n 个不相邻的点，于是答案显然是 $2^{\text{环数}}$，直接并查集即可。

T2

首先若存在 $a_i = i$，显然无解。

若 $a_i > i$，则我们需要把这个数字从 i 位置向右挪到 a_i 位置上。于是就会发现相邻位置的交换顺序有一些限制，限制形如某对相邻的交换必须在它旁边的相邻对交换之前/之后。

$a_i < i$ 就是把 i 位置向左挪到 a_i 的位置上，限制也类似。

于是问题就变成了有若干个形如 $b_i > b_{i-1}$ 或 $b_i < b_{i-1}$ 的限制，问满足要求的排列 b 有多少个。

直接 $O(n^2)$ dp 即可，当然可以容斥+分治 FFT 优化成 $O(n\log^2 n)$，但由于这是 NOIP 模拟赛就不说了。

T3

事实上只需要把前两行和前两列变成 0 即可，如果此时还不能把整个矩阵归零那么必然无解。证明就考虑任意一个 3×3 的矩阵，$a_{1,1}-a_{1,2}-a_{2,1}+a_{2,3}+a_{3,2}-a_{3,3}$ 的值在这些操作下永远不变。

于是这样的构造是很容易的，4000 次操作就足够了。

T4

简单的斜率优化。

考虑 $f_{i,j}$ 表示当前 dp 到了 i，上一个区间右端点为 j 时的最优答案。则显然有：

$$f_{i,j} = \max_{0\le k<j} f_{j,k} + (s_i - s_j)(s_j - s_k)$$

显然是个斜率优化的形式，扔掉只和 i,j 有关的常数项，则可以写成：

$$f_{i,j} = p + \max_{0\le k<j} f_{j,k} - t\cdot s_k$$

若对于 $s_a < s_b$ 有 $f_{j,a} - t\cdot s_a < f_{j,b} - t\cdot s_b$，那么显然：

$$\frac{f_{j,a}-f_{j,b}}{s_a-s_b}>t$$

于是我们枚举 j，按照 s 排序后暴力维护出一个斜率递减的上凸壳。注意到 $t=s_i-s_j$，我们再按照 s_i 从大到小枚举 i，就可以贪心地从凸包前面删点了。

于是总复杂度 $O(n^2)$。