

签到题

不断将 n 除以 60，将余数求和即可。注意到 $60^{11} > 10^{18}$ ，因此在 11 次除法内 n 一定会归零。

过了个过

我们将串分成若干极长的 01 交替的串，那么我们每次可以取这些串中的一个，但每次取完都可能会有相邻的极长 01 串合并。

观察合并的过程，仅有在取走一个左边被 0 阻隔，右边被 1 阻隔或是左边被 1 阻隔右边被 0 阻隔的串，我们才会进行合并，而且我们一定会去贪心地选择这样的串移去。

我们数出全串 00 的间隔数量 c_0 ，11 的间隔数量 c_1 ，那么我们一定可以做到选择 $\min(c_0, c_1)$ 个这样的串去除（一旦两种间隔都存在，就必定存在相邻的不同间隔），于是答案为 $c_0 + c_1 + 1 - \min(c_0, c_1)$ 。

河了个河

首先可以通过归并排序求出序列的逆序对数量 c 。

我们考虑循环移位操作对逆序对数量的影响：将 a_1 由开头移至结尾，所有小于其的数都会由逆序对变为顺序对，而所有大于其的数都会由顺序对变为逆序对，因此逆序对数量由 c 变为 $c - (a_1 - 1) + (n - a_1)$ 。

而翻转更容易刻画，所有逆序对变为顺序对，顺序对变为逆序对，于是逆序对数量由 c 变为 $\frac{n(n-1)}{2} - c$ 。

我们只需维护目前在操作哪一个数，复杂度 $O(n \log n + q)$ 。

卒了个卒

先考虑如何模拟判定过程：从前往后扫描数轴上每个位置，我们维护两个 bit u, v ，分别表示当前有没有球往后滚，以及有球的洞数量奇偶性。

若我们遇到一个球，就会将 u 置为 1；若我们遇到一个洞，就会将 v 异或上 u ，并将 u 置为 0。

对 bit 的操作启示我们使用 bitset 加速模拟，枚举 S_j 使用两个长为 n 的 bitset 维护 u, v ，从前往后扫描每个位置，并对 bitset 进行相应的位运算即可。

复杂度 $O(\frac{n^2 m}{w})$ ，其中 $w = 64$ 。