考虑一个 1010101 形状的子串，它是无法被操作的，因此最终的串形态一定是这样 01 交替。

我们称这样的（无法被操作的）串是稳定的。

结论：操作可以看成每次取 SS^T 做操作，其中 S 是稳定的。

证明：否则你总可以先对 S 做操作，直到其稳定。

于是把原串分成很多个长成这样的稳定子串，那么操作可以看作取相邻的两个子串做抵消。

例如：

$$\begin{aligned}&1010101101000101\\ \to\ &(1010101)(101)000(101)\end{aligned}$$

设一个稳定串的权值是其中 1 的数目，你的目标是使最后留下的稳定串的权值最小。

你的操作是每次取两个相邻的稳定串，使它们的权值共同减小一个正整数。

那么设 S 中串的权值是 $a_1,\cdots,a_n$。

根据经典结论，如果 $a_1\ge\sum_{i\ge2}a_i$，最后会剩下一个权值为 $a_1-(\sum_{i\ge2}a_i)$ 的串。

否则剩下一个权值为 $(\sum a_i)\bmod 2$ 的串。

这样我们就能知道一个串最终会剩下多少个 1。

只用考虑最终的串的左右侧是否会留下 0。

先排除没有 1 的情况。

如果最初的串不是一个整体（指并非一次都操作不了），那就不可能左右两侧一个 0 都没有。

只用考虑何时左右两侧都有 0。

讨论：

1. 最初两侧都是 0

   那无论如何两侧都会有 0。
2. 最初一侧是 1

   考虑保住这一边。

   当 a_1 不是这一侧，且 $a_1>\sum_{i\ge2}a_i$ 时就保不住。否则总能保住。
3. 两侧是 1

   无非和上面情况类似。

   a_1 不是两侧，且 $a_1>\sum_{i\ge2}a_i$ 时就保不住。

总复杂度 $O(n\log n)$ 可以通过此题。