1.2 $30pts$ 做法

显然，旋转后的序列只有 n 种。我们直接枚举，每次 $O(n)$ 计算。取最小值即可。

时间复杂度 $O(n^2)$。

1.3 $70pts$ 做法

记旋转 i 次后的健美值为 ans_i。我们单独考虑序列中的每一个元素。

对于所有 $i \in [1, n]$，若 $s_i < i$，则：

(1) 对 $ans_0, ans_1, \cdots, ans_{i-s_i-1}, ans_{i-s_i}$ 分别产生 $i-s_i, i-s_i-1, \cdots, 1, 0$ 的贡献。

(2) 对 $ans_{i-s_i+1}, ans_{i-s_i+2}, \cdots, ans_{i-2}, ans_{i-1}$ 分别产生 $1, 2, \cdots, s_i-2, s_i-1$ 的贡献。

(3) 对 $ans_i, ans_i+1, \cdots, ans_{n-2}, ans_{n-1}$ 分别产生 $n-s_i, n-s_i-1, \cdots, i-s_i+2, i-s_i+1$ 的贡献。

$s_i > i$ 及 $s_i = i$ 时处理类似。

容易发现，所产生的贡献均为一段连续的上升（下降）区间，我们只需要考虑实现一个支持区间递增加（减）的算法即可。

一个直观的做法：线段树。

每个节点维护三个 tag，分别代表区间加，递增（减）加。然后在叶子节点维护权值。注意下放递增（减）加 tag 时要同时打上区间加 tag。大力调试即可，时间复杂度 $O(n \cdot logn)$。

1.4 $100pts$ 做法

事实上，在这里用线段树显得很浪费，因为我们并不需要在线查询 ans 的值，所以有更优秀的算法来解决这一问题。

首先简单回顾一下如何快速对序列进行区间加：

假如要对数组 a 在区间 $[l, r]$ 上加上 w，记 $a_n = \sum_{i=1}^{n} a'_i$，那么我们只需把 a'_l 加上 w，把 a'_{r+1} 减去 w，最后统计 a' 的前缀和即可。

对于本题，我们可以将区间 $[l, r]$ 递增加拆开成把区间 $[l, r]$ 加上一个常数，再分别加上 $1, 2, \cdots, r-l+1$。前者我们直接应用上面的方法。对于后者，我们设辅助数组 d, f，记 $d_n = \sum_{i-1}^{n} f_i$，然后我们将 $f_{l..r}$ 加上 1，将 f_{r+1} 减去 $(r-l+1)$。最后统计答案时将 ans_i 加上 d_i 即可。区间递减加的处理类似。

这样每次修改是 $O(1)$ 的，最后统计答案是 $O(n)$ 的。总时间复杂度为 $O(n)$。