# T1-football

对于一支队伍而言，首先其必然要获胜，它的最终分数也已经确定了。

对于一次比赛的两只队伍而言，不妨设其初始分数从小到大为 $x, y$。设某个队伍的最终分数为 $k$。

- 当 $k < x$ 时，排名必然上升两个；
- 当 $x + 3 \leq y$ 时，$k < y$ 则排名上升一个，否则不上升；
- 其它情况下，$k \leq y$ 则排名上升一个，否则不上升。

上面这些可以枚举一下胜负状态得到。

随后，可以先算所有比赛的排名上升状态，然后再减去有关自己比赛的。可能有一些细节。

# T2 - sakuzyo

首先观察到只有形如 `yyyyy...yyylqqq...qqqqq` 的子串才有用，不妨先将其摘出来，连带 `l` 位置的奇偶性。

如果删除了 `l`，那么这一部分就完全没办法继续删了。我们希望 `l` 位置的奇偶性可以不断被改变。同时，也只有删除了其前面的 `l`，才能改变当前的奇偶性。

所以应该让每一段都尽量删一次 `l`，在此之前删除尽可能多的 `y` 和 `q`。如果从前往后看的话，可以直接维护之前删了多少个 `l`，模拟一下就好了。可能需要一些分类讨论。

# T3 - vex

发现只有上下飞要代价，也就是说只用看两个点间路径最小值最大是多少。

可以建最大瓶颈树然后做链求最小值，不过可能常数大一点。

std 是离线下来，把询问挂到点上，从大到小加点做启发式合并。

# T4 - modulo

首先注意到 $x$ 取模之后，要么值不变，要么值至少减半。设 $V$ 是值域。

## 解法 1

先把 $a_i$ 找到一个前缀最小值序列，只有这些位置的值可能有用。

考虑如何计算一个值域前缀 $x$ 的答案：找到第一个小于等于它的 $a_i$，分解成一些 $a_i - 1$ 和一个 $x \bmod a_i$。这启示我们从后往前计算不断 $a_i - 1$ 的答案，每个只要 $O(\log n \log V)$ 的复杂度。

对于查询，如上所述，进行一次也是 $O(\log n \log V)$ 的。

## 解法 2

从前往后枚举时，用优先队列维护现在要求的一些前缀 和它们的系数。每次新来一个 $a_i$，就不断弹出前缀位置大于等于 $a_i$ 的，设为 $x$，将其分解成一些 $a_i - 1$ 和 $x \bmod a_i$。然后将所有的前缀 $a_i - 1$ 的系数求和合并起来，这样被加入的非取模后的数只有 $O(n)$ 个，而它们每个最多只会被取模 $O(\log V)$ 次，总的复杂度就是 $O(n \log n \log V)$。