

2026 CCF CAT 省赛题解

Coordinated by: jiangdu & frost_ice

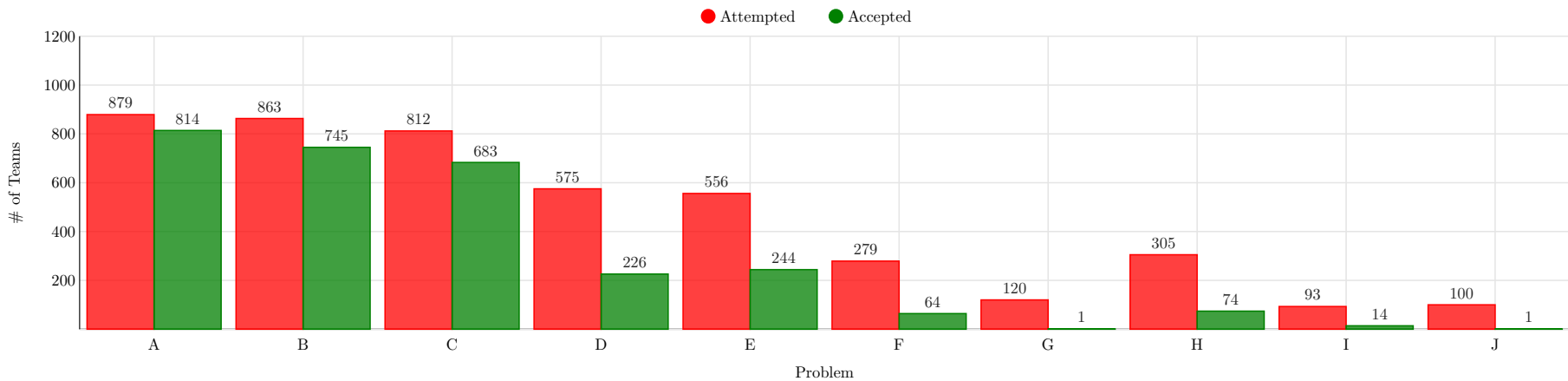
2026-06-14

哈尔滨工程大学

概览

预期难度:

题号	A	B	C	D	E	F	G	H	I	J
预期难度	very easy			easy		mid			hard	
实际通过	814	745	683	226	244	64	1	74	14	1



概览

A

B

C

D

E

F

G

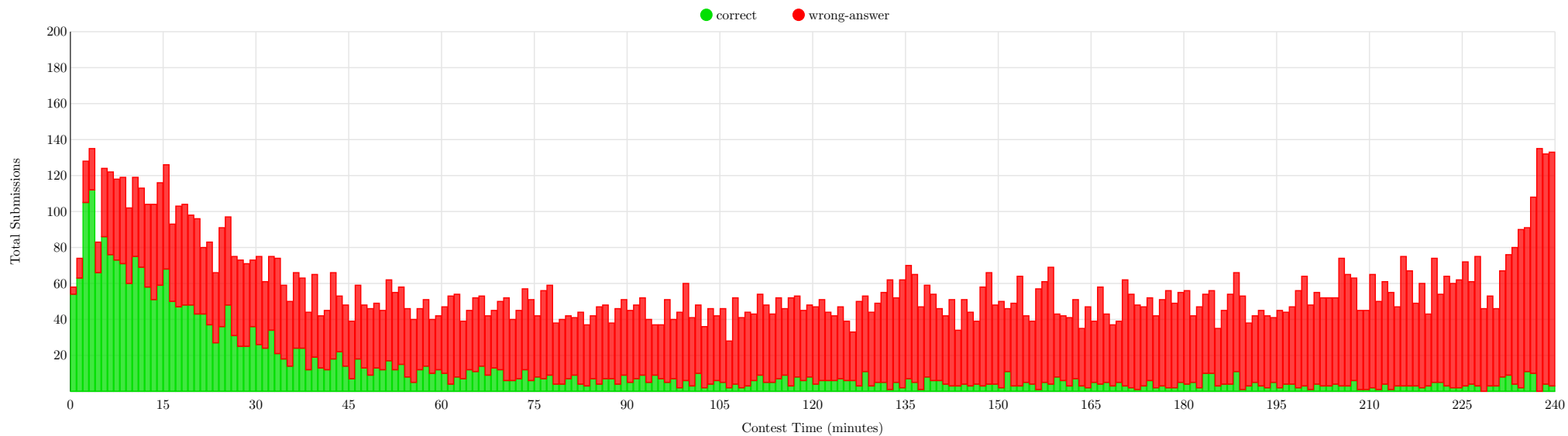
H

I

J



概览



A. 供电系统

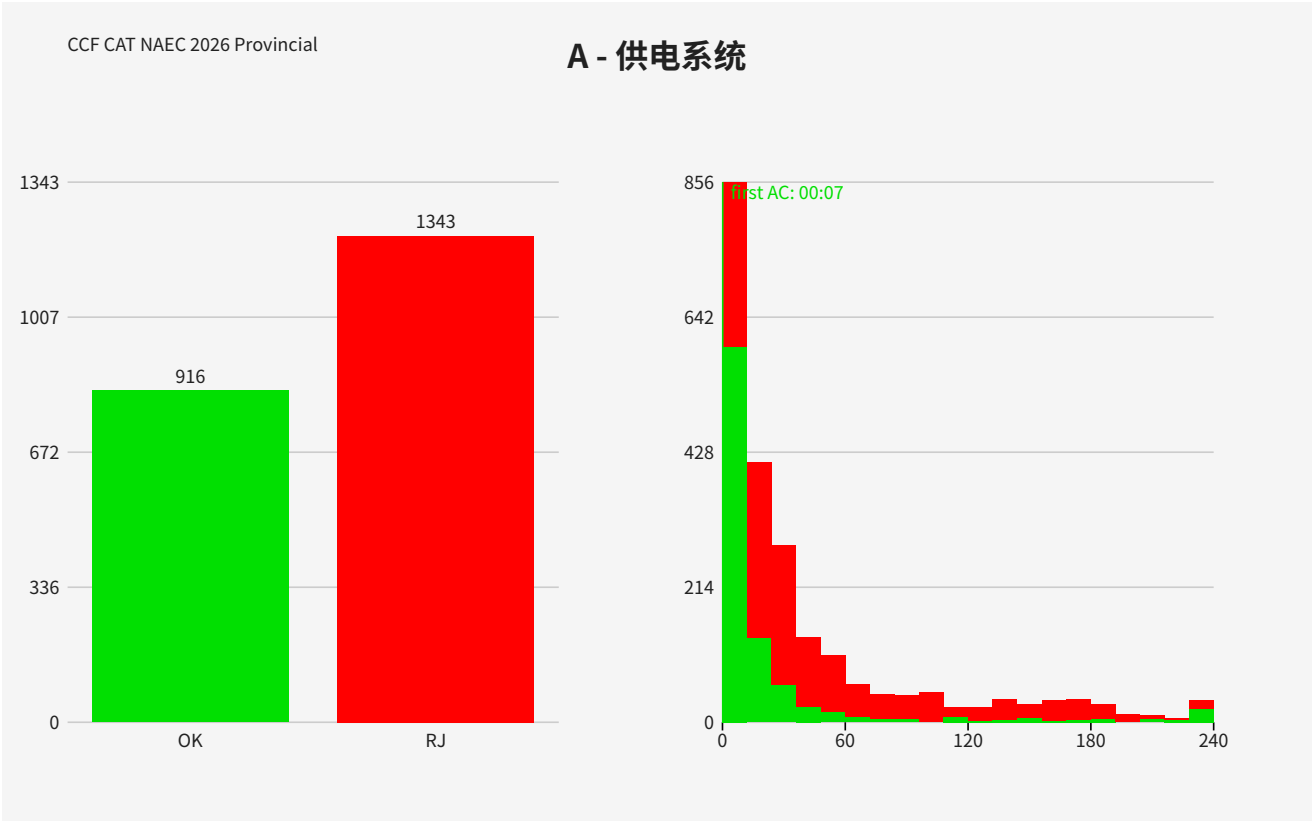
题意

- 有 $10^9 + 1$ 个供电节点，分别编号为 0 到 10^9 ；
- 给定整数 X, Y, Z ($0 \leq X, Y, Z \leq 10^9$, $X \leq Y$)；
- 已知 $[X, Y]$ 区间内的节点处于「有电」状态，其余节点处于「没电」状态；
- 询问 Z 节点是否处于「有电」状态；

按照题意分类讨论即可：

- 若 $X \leq Z \leq Y$ (即 Z 位于区间 $[X, Y]$ 中)，输出 YES；
- 否则，输出 NO；

A. 供电系统

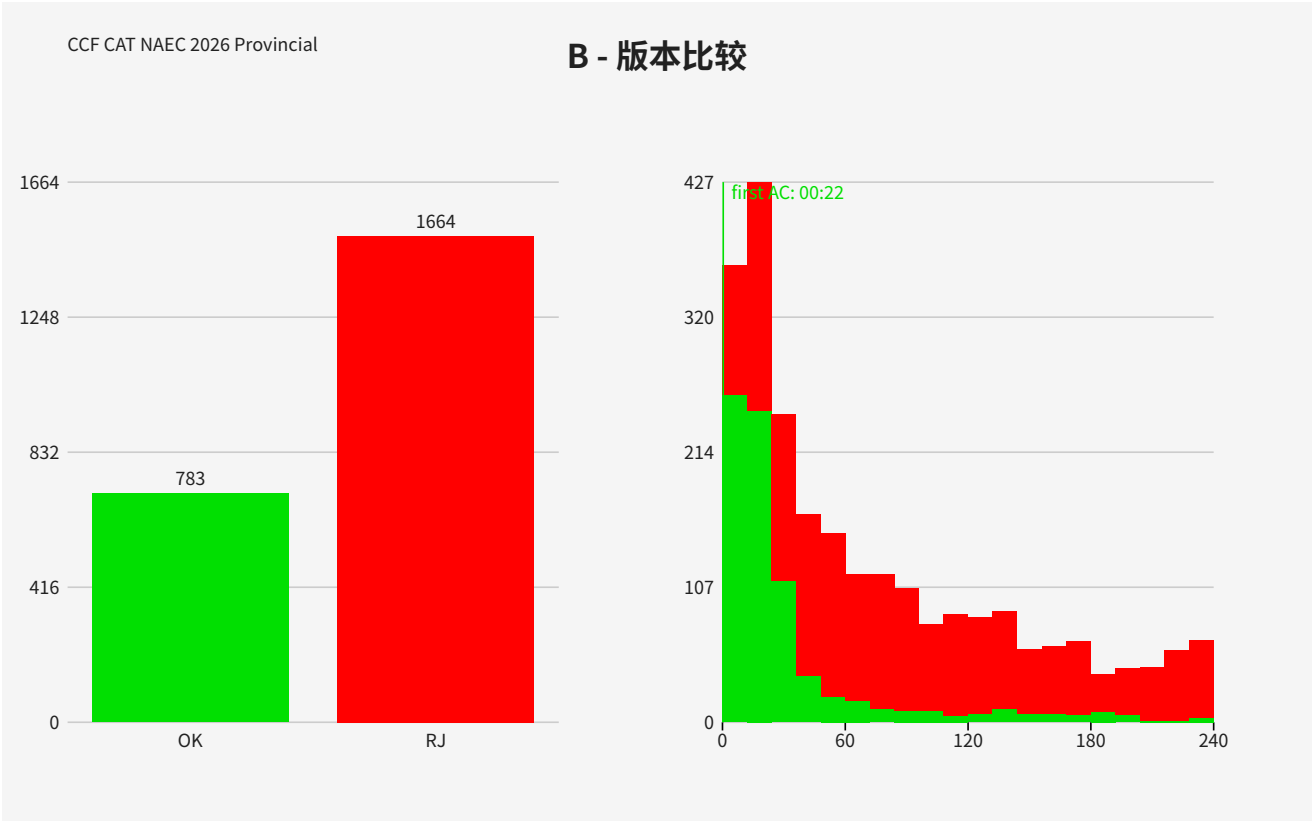


B. 版本比较

题意

- 给定两个版本号 $X_1.Y_1.Z_1$ 以及 $X_2.Y_2.Z_2$;
 - 要求比较两个版本号的大小关系;
-
- 按照题目给出的比较规则, 分类讨论即可;
 - 注意到比较规则实际是在比较序列 $[X_1, Y_1, Z_1]$ 与 $[X_2, Y_2, Z_2]$ 的**字典序**大小关系;
 - 也可以借助你使用的编程语言的相关标准库工具实现, 如 c++ 的 `std::array`, python 的 `tuple` 等, 它们默认实现的比较函数实际就是在比较字典序;

B. 版本比较

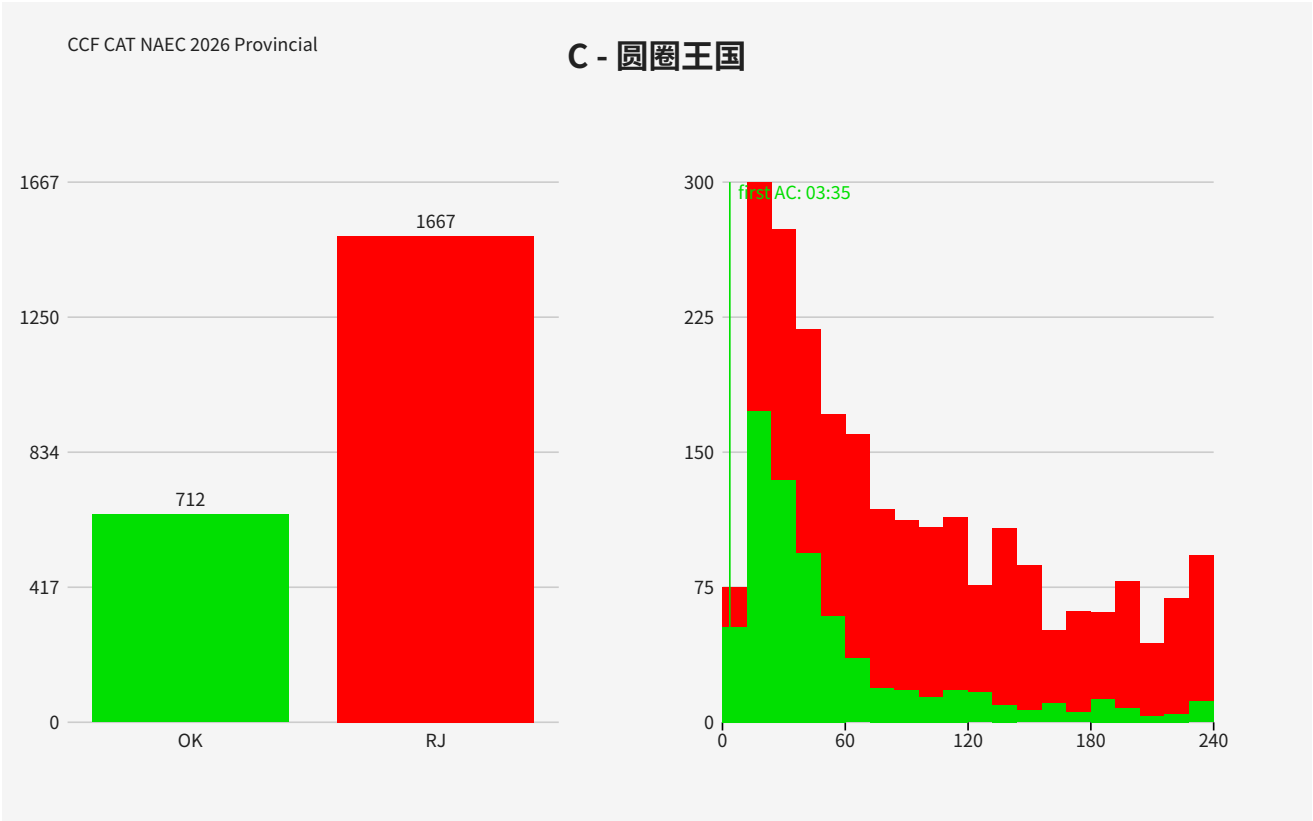


C. 圆圈王国

题意

- 给定一个非负整数 x ，要求你将 x 转换成一种特殊的三进制表示：
 - 位权低的在前，高的在后（与正常进制相反）；
 - 使用 数字 0、小写字母 o、大写字母 O 这三种字符分别表示数字 0、1、2；
 - 相邻两位之间使用字符 . 分隔；
- 单测， $0 \leq x \leq 10^9$ ；

C. 圆圈王国



C. 圆圈王国

按题意模拟即可：

- 对于 x 是 0 的情况：
 - 直接输出 0 并结束；
- 否则，循环执行以下操作，直到 x 变为 0：
 - 如果不是第一次进入循环，输出 .；
 - 计算 $x \bmod 3$ 得到当前的最低位，并按它的值是 0 / 1 / 2，输出 0 / o / 0；
 - 将 x 赋值为 $\lfloor \frac{x}{3} \rfloor$ ，去掉当前的最低位；

时间复杂度 $O(\log_3 x)$

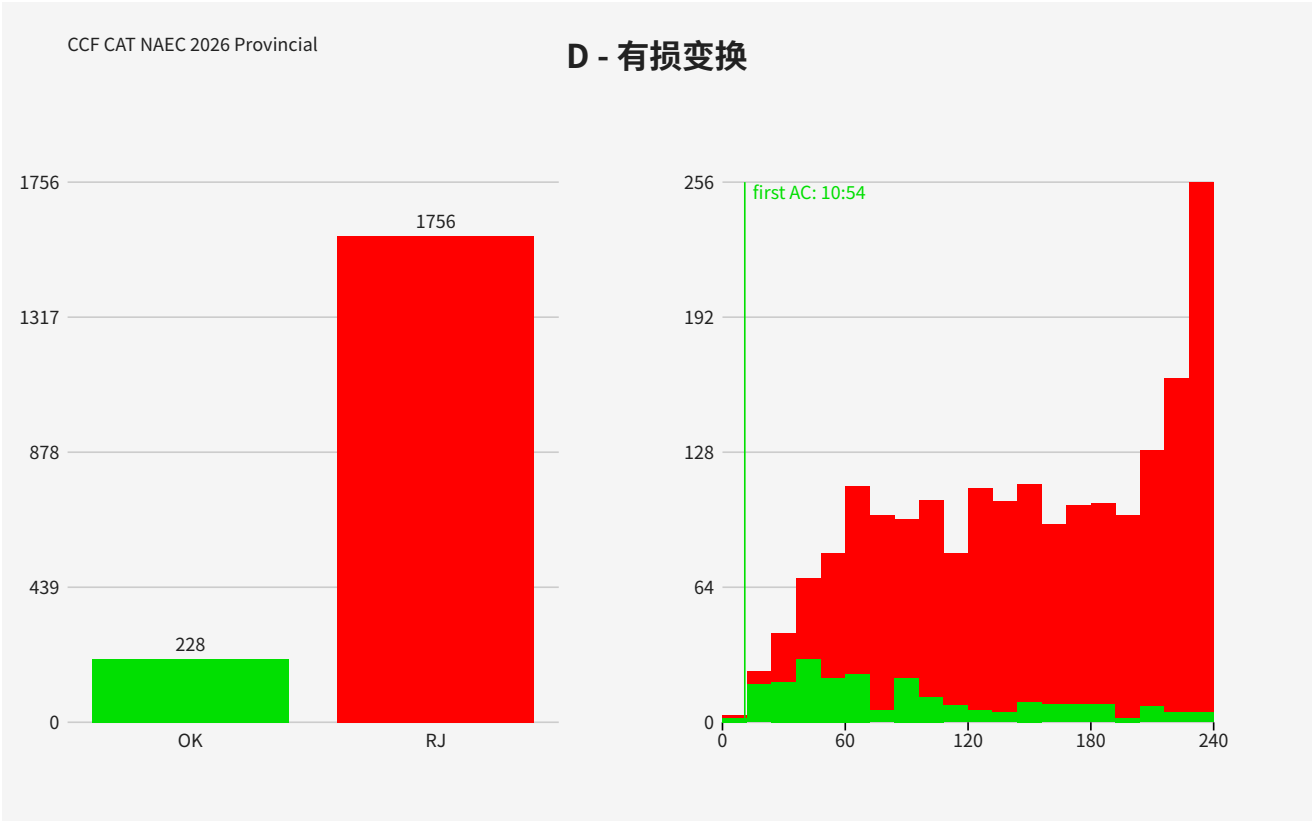
D. 有损变换

Prepared by: frost_ice

题意

- 给定一个整数 y ，要求你寻找一个整数 x ，使得 x 和 $2x$ 的按位或是 y ；
- 判断是否有解，并给出构造；
- $0 \leq y < 2^{60}$ ，多测， $T \leq 10^4$ ；

D. 有损变换



D. 有损变换

为了避免混淆，我们接下来用 $\text{or}(a, b)$ 表示 a 和 b 的逻辑或， $\text{bitand}(a, b)$ 表示 a 和 b 的按位与；

假设一个数 y 能够通过 x 和 $2x$ 按位或得到，我们考虑 y 有什么特殊性质；

- 令 x 的二进制表示是 $\overline{\dots a_2 a_1 a_0}$ ， y 的二进制表示是 $\overline{\dots b_2 b_1 b_0}$ （从高位到低位）；
- 特别的，对于负整数 i ，我们规定 a_i 和 b_i 为 0；
- 由于 y 是 x 和 $2x$ 的按位或，根据定义，对于整数 i ，有：

$$b_i = \text{or}(a_i, a_{i-1})$$

D. 有损变换

引理 1

- 对整数 i , 若 b_i 为 1, 则 b_{i-1} 和 b_{i+1} 至少有一个也为 1;

由于 $b_i = \text{or}(a_i, a_{i-1}) = 1$, 因此「 a_i 为 1」和「 a_{i-1} 为 1」至少有一个成立。

对于这两种情况:

- 当 a_{i-1} 为 1 时, $b_{i-1} = \text{or}(a_{i-1}, a_{i-2}) = \text{or}(1, a_{i-2}) = 1$;
- 当 a_i 为 1 时, $b_{i+1} = \text{or}(a_{i+1}, a_i) = \text{or}(a_{i+1}, 1) = 1$;

因此「 b_{i-1} 为 1」和「 b_{i+1} 为 1」至少有一个成立, 引理得证;

D. 有损变换

引理 1 直观的理解是：

- 假设一个数 y 能够通过 x 和 $2x$ 按位或得到，那么它的二进制表示中，不存在孤立的 1；

也就是说：

- 对于输入的数 y ，只要它的二进制表示中，存在一个孤立的 1，则一定无解；
- 对于剩下下来的情况，所有极长的连续的 1 组成的段，长度一定 ≥ 2 ；
- 接下来考虑这种情况下如何构造答案；

D. 有损变换

- 不难发现，对于每个极长的连续的 1 组成的段，我们去掉其中位最高的 1，即可得到一个满足条件的 x ，例如：

$$\begin{aligned}
 x &= \overline{\dots 00\textcolor{red}{0}110000\textcolor{red}{0}111001} \\
 2x &= \overline{\dots 00\textcolor{blue}{1}1\textcolor{red}{0}00000\textcolor{blue}{1}11001\textcolor{red}{0}} \\
 y = \text{bitor}(x, 2x) &= \overline{\dots 00\textcolor{blue}{1}110000\textcolor{blue}{1}111011}
 \end{aligned}$$

- 实现上，可以遍历 y 的每个二进制位并处理。时间复杂度 $O(w)$ （其中 $w = 60$ 表示二进制位数）；
- 或者直接令 x 为 $\text{bitand}(y, \lfloor \frac{y}{2} \rfloor)$ （即去掉每个极长的连续的 1 组成的段的最高位的 1），并判断 $\text{bitor}(x, 2x)$ 是否等于 y 。时间复杂度 $O(1)$ ；

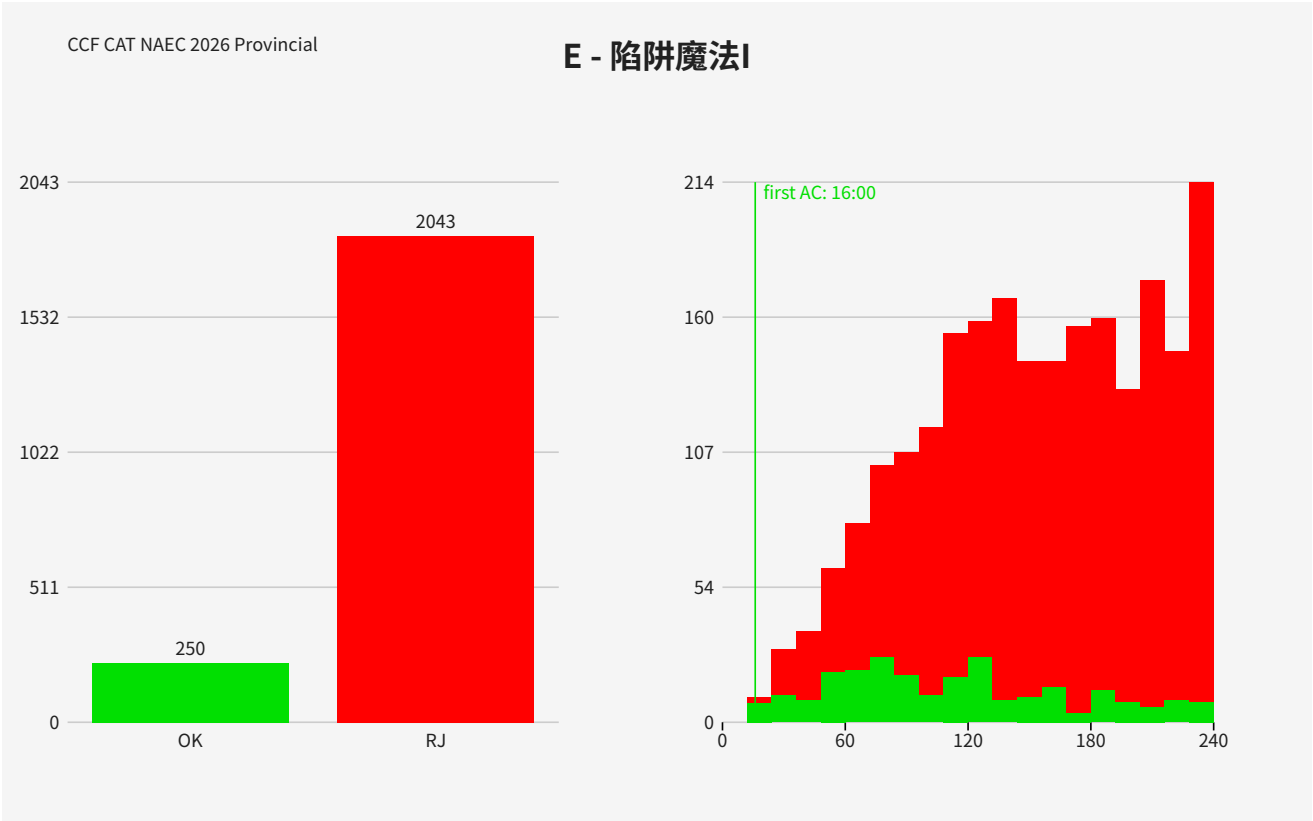
E. 陷阱魔法 I

Prepared by: alwaysBelieve

题意

- 给定一个长为 n 的排列 $[a_1, a_2, \dots, a_n]$;
- 要求选出一个尽可能小的非负整数 x , 使得:
 - 通过若干次【将一个「最大值和最小值之差不超过 x 」的区间内的数任意重排】的操作, 可以将原序列排成从小到大的顺序;
- 多测, $\sum n \leq 10^5$;

E. 陷阱魔法 I



E. 陷阱魔法 I

我们首先证明下界。

引理 1 $x \geq \max\{a_i - a_j \mid i, j \in \{1, 2, \dots, n\} \wedge i < j\}$

- 考虑形如 $a_i > a_j$ 且 $i < j$ 的逆序对。
- 初始时，数字 a_i 在数字 a_j 的前面；当整个序列有序之后，数字 a_i 需要出现在数字 a_j 的后面（否则序列就不是有序的）；
- 因此，在排序的过程中，必定至少涉及到一次，改变这两个数字的相对位置关系。
- 这样一来，对于任意的 $a_i > a_j$ 且 $i < j$ 的逆序对， x 都不能小于 $a_i - a_j$ ，否则这两个数字永远没有机会交换相对位置关系；

E. 陷阱魔法 I

接下来我们证明上界也是它。考虑使用类似**插入排序**的思路进行数学归纳。

引理 2 取 $x = \max\{a_i - a_j \mid i, j \in \{1, 2, \dots, n\} \wedge i \leq j\}$, 对于任意 $1 \leq k \leq n$, 存在一种将子区间 $[1, k]$ 排成有序的操作方案, 并且所有操作只影响前 k 个数。

首先证明初始情况。 $k = 1$ 时显然成立;

接下来考虑, 所有操作只影响前 k 个数, 并且子区间 $[1, k]$ 已经有序的情况下, 如何通过操作, 使得所有操作只影响前 $k + 1$ 个数, 并且子区间 $[1, k + 1]$ 也被排成有序的;

E. 陷阱魔法 I

- 此时前 k 个数已经有序, 因此它必然可以分成前半段比 a_{k+1} 小的数, 与后半段比 a_{k+1} 大的数; 我们的目标是不改变这些数的相对顺序, 并将 a_{k+1} 放置在这两种数的中间;
- 假设区间 $[1, m)$ 内的数都 $< a_{k+1}$, 区间 $[m, k+1)$ 内的数都 $> a_{k+1}$;
- 我们直接选取区间 $[m, k+1]$ 应用操作。可以证明, 此时最大值和最小值之差一定不超过 x :

$$\begin{aligned}
 \left(\max_{m \leq i \leq k+1} a_i \right) - \left(\min_{m \leq i \leq k+1} a_i \right) &= \max\{a_i - a_{k+1} \mid i \in \{m, m+1, \dots, k+1\}\} \\
 &\leq \max\{a_i - a_j \mid i, j \in \{1, 2, \dots, n\} \wedge i \leq j\} \\
 &= x
 \end{aligned}$$

- 操作结束后, 子区间 $[1, k+1]$ 即被排成有序。引理得证。

E. 陷阱魔法 I

- 因此我们的目标是计算出 $\max\{a_i - a_j \mid i, j \in \{1, 2, \dots, n\} \wedge i \leq j\}$ 。
- 考虑从前往后遍历每一个数并维护答案：
 - 将变量 i 从 1 枚举到 n ，实时维护出前 i 个数的最大值（记为 pre_i ），并使用 $\text{pre}_i - a_i$ 更新答案即可。
- 时间复杂度 $O(n)$ 。
- 由于可以边读入边处理，实时维护 $O(1)$ 个变量，空间复杂度可以做到 $O(1)$ 。

F. 奇怪染料

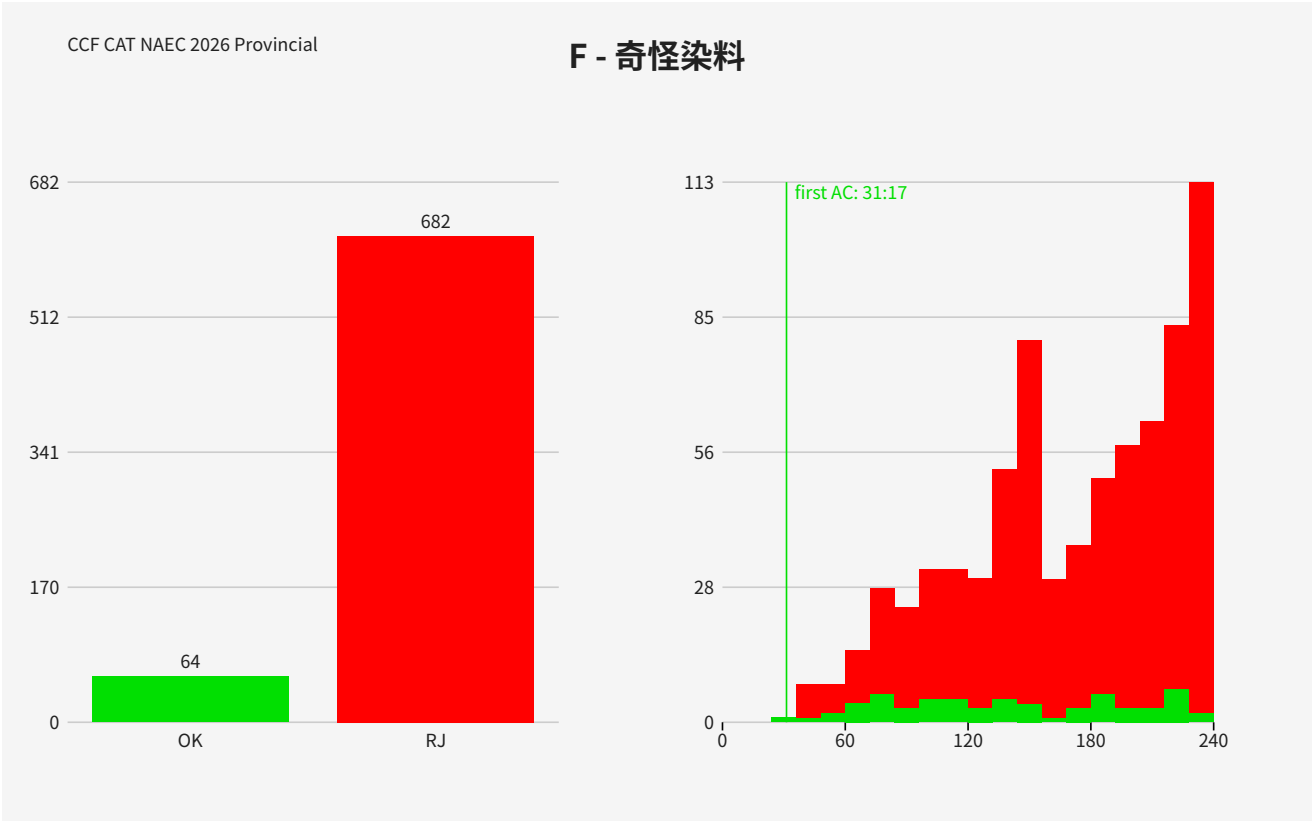
Prepared by: Sakura__ylr

题意

- 给定一个长度为 n 的由 R（红） G（绿） B（蓝） 三种颜色以及 N（空白）组成的目标序列。
- 初始时，有一个长度为 n 的序列，所有位置均为 N（空白）。每次操作，可以选择一个区间 $[l, r]$ 和一种颜色 c ，对区间进行染色操作。染色的规则如下：
 - 原先是 N（空白）的位置，刷上颜色 c ，变为颜色 c ；
 - 原先是颜色 c 的位置，再刷一次颜色 c ，则无变化；

- 原先是颜色 x 的位置，刷上颜色 y ($x \neq y$)，则会变成颜色 z ，满足 $z \neq x$ 且 $z \neq y$ 。
- 询问最少需要几次染色操作才能得到给定的目标序列；
- 多测， $\sum n \leq 2 \times 10^5$ 。

F. 奇怪染料



F. 奇怪染料

引理 1

- 整个序列被 N 自然分割成若干个只由 R、G、B 组成的「极长连续彩色块」，每个彩色块可以独立求解，总操作次数即为所有彩色块的最小操作次数之和；
- 一旦某个位置被染上颜色 (R、G 或 B)，由于反应规则中没有任何一条能让颜色变回 N ，该位置将永久保持有色状态；
- 因此，任何染色操作的区间 $[l, r]$ 绝不能覆盖目标序列中为 N 的位置；
- 因此，对于目标序列中，两个被 N 分隔开的位置，不存在操作可以同时覆盖到这两个位置，因此它们一定是互不相关的；

F. 奇怪染料

引理 2

- 对于连续出现的相同颜色，将其压缩成单个字符（例如 RRRGGGB 可以压缩为 RGB），不会改变最小操作次数。

在最优的操作方案中，我们一定不会将一次操作的端点，插在一段连续的相同的颜色中间；否则，我们一定可以通过调整，并让结果变得不劣。具体的：

- 假设目标序列中， $[L, R]$ 段的颜色相同。我们想办法将 $[L, L]$ 被操作的情况应用于整个 $[L, R]$ 区间。对于操作 $[l_i, r_i]$ ：

F. 奇怪染料

- 如果 $[L, R] \subset [l_i, r_i]$ 或者 $[L, R] \cap [l_i, r_i] = \emptyset$ ，即它原先已经操作了整个区间，或者与区间没有关系，则不做调整；
- 否则，若 $[l_i, r_i] \cap \{L\} \neq \emptyset$ ，即它原先能够操作到 $[L, L]$ ，将区间调整为 $[l_i, r_i] \cup [L, R]$ ，即可将操作应用于整个区间；
- 否则，若 $[l_i, r_i] \subset [L + 1, R]$ ，即它原先只能操作到 $[L + 1, R]$ 里面的范围，则删去这次操作；
- 否则，调整为 $[l_i, r_i] \cap [R + 1, n]$ ，即去掉这次操作在 $[L + 1, R]$ 中的部分；
- 显然，经过调整，操作次数一定不会变得更劣，并且再也没有操作的端点是插在区间 $[L, R]$ 中的。

F. 奇怪染料

我们首先证明答案的下界。

引理 3

- 对于一个长度为 L 的序列 $[t_1, t_2, \dots, t_L]$, 满足 $t_i \in \{R, G, B\}$ 且 $t_i \neq t_{i+1}$, 至少需要 $\lceil \frac{L+1}{2} \rceil$ 次操作才能完成染色;

我们将 $t_i \neq t_{i+1}$ 的相邻位置对 $(i, i + 1)$ 称为分界点。特别的, 我们将 $(0, 1)$ 和 $(L, L + 1)$ 也看成是分界点。

直观的理解是, 如果没有任何一次操作的区间的端点落在这两个位置之间 (即没有操作的区间满足 $l = i + 1$ 或 $r = i$), 那么染色完成后, i 和 $i + 1$ 位置的颜色一定是相同的。

F. 奇怪染料

下面给出详细证明。

- 目标序列中共有 $L + 1$ 个分界点。初始状态下全部为 N ，分界点数量为 0。
- 假设某一步我们选择区间 $[l, r]$ 并涂上颜色 c 。对于区间内部的任意相邻位置对 $(i, i + 1)$ ($l \leq i < r$)，若它们在染色前状态相同，不难证明染色完成后状态依然相同。说明区间内部不可能产生新的分界点。
- 对于区间外部的相邻位置对，颜色未变，不可能产生新的分界点。
- 唯一可能产生新的分界点的位置在区间的边界上，即 $(l - 1, l)$ 与 $(r, r + 1)$ 。
- 意味着，每次操作，最多新增 2 个分界点。
- 因此，要产生 $L + 1$ 个分界点，至少需要进行 $\lceil \frac{L+1}{2} \rceil$ 次操作。

F. 奇怪染料

接下来我们证明上界也刚好是它。

引理 4

- 对于任意一个长度为 L 的序列 $[t_1, t_2, \dots, t_L]$, 满足 $t_i \in \{R, G, B\}$ 且 $t_i \neq t_{i+1}$, 均可在 $\lceil \frac{L+1}{2} \rceil$ 次操作下构造出来。

接下来我们通过数学归纳证明。首先证明起始情况。

- $L = 1$ 的情况, 通过 1 次操作, 在区间 $[1, 1]$ 上涂上颜色 t_1 即可。
- $L = 2$ 的情况, 通过 2 次操作, 在区间 $[1, 1]$ 和 $[2, 2]$ 上分别涂上颜色 t_1 和 t_2 即可。

F. 奇怪染料

假设对于所有长度为 $L \leq k$ ($k \geq 2$) 的序列，引理 4 均成立。现在我们来证明长度为 $L = k + 1$ 的序列，引理 4 也成立。我们分奇偶进行讨论。

若 L 为偶数，不妨令 $L = 2m$ 。我们的目标是证明它可以通过 $m + 1$ 次操作构造出来。

- 我们将序列分成前缀 $[t_1, t_2, \dots, t_{L-1}]$ 以及后缀 $[t_L]$ 两部分。根据归纳假设，前者可以通过 $\lceil \frac{L-1+1}{2} \rceil = \frac{2m}{2} = m$ 次操作构造出来，后者可以通过 1 次操作构造出来，因此分别构造即可。

L 为偶数的情况得证。

F. 奇怪染料

若 L 为奇数, 不妨令 $L = 2m + 1$ 。我们的目标是证明它可以通过 $m + 1$ 次操作构造出来。

我们分类讨论 $[t_{L-2}, t_{L-1}, t_L]$ 这三个数的情况。我们已知的条件是 $t_{L-1} \neq t_L$ 以及 $t_{L-2} \neq t_{L-1}$, 因此我们只需要讨论 t_{L-2} 究竟是等于 t_L , 还是等于 t_L 和 t_{L-1} 之外的第三种颜色。

- 情况 1: $[t_{L-2}, t_{L-1}, t_L] = [x, y, z]$, 其中 x 、 y 、 z 互不相等:
 - 根据归纳假设, 序列 $[t_1, t_2, \dots, t_{L-3}, x, \mathbf{N}, \mathbf{N}]$ 可以通过 m 次操作构造出来。显然, 同样的次数下, 也可以构造出 $[t_1, t_2, \dots, t_{L-3}, x, x, \mathbf{N}]$ 。
 - 接下来将 $[L - 1, L]$ 区间染上颜色 z , L 位置变为颜色 z , $L - 1$ 位置由于反应, 变为颜色 y 。
 - 这样一来, 我们用 $m + 1$ 次操作得到了 $[t_1, t_2, \dots, t_{L-3}, x, y, z]$;

F. 奇怪染料

- 情况 2: $[t_{L-2}, t_{L-1}, t_L] = [z, y, z]$, 其中 $y \neq z$:
 - 我们用 x 表示与 y 、 z 都不相等的颜色;
 - 根据归纳假设, 序列 $[t_1, t_2, \dots, t_{L-3}, z, \mathbf{N}, \mathbf{N}]$ 可以通过 m 次操作构造出来。显然, 同样的次数下, 也可以构造出 $[t_1, t_2, \dots, t_{L-3}, z, z, z]$ 。
 - 接下来将 $[L-1, L-1]$ 区间染上颜色 x , $L-1$ 位置由于反应, 变为颜色 y 。
 - 这样一来, 我们用 $m+1$ 次操作得到了 $[t_1, t_2, \dots, t_{L-3}, z, y, z]$;

L 为奇数的情况得证;

综上, 我们证明了, 假设对于所有长度为 $L \leq k$ ($k \geq 2$) 的序列, 引理 4 均成立。那么对于长度为 $L = k + 1$ 的序列, 引理 4 也成立。

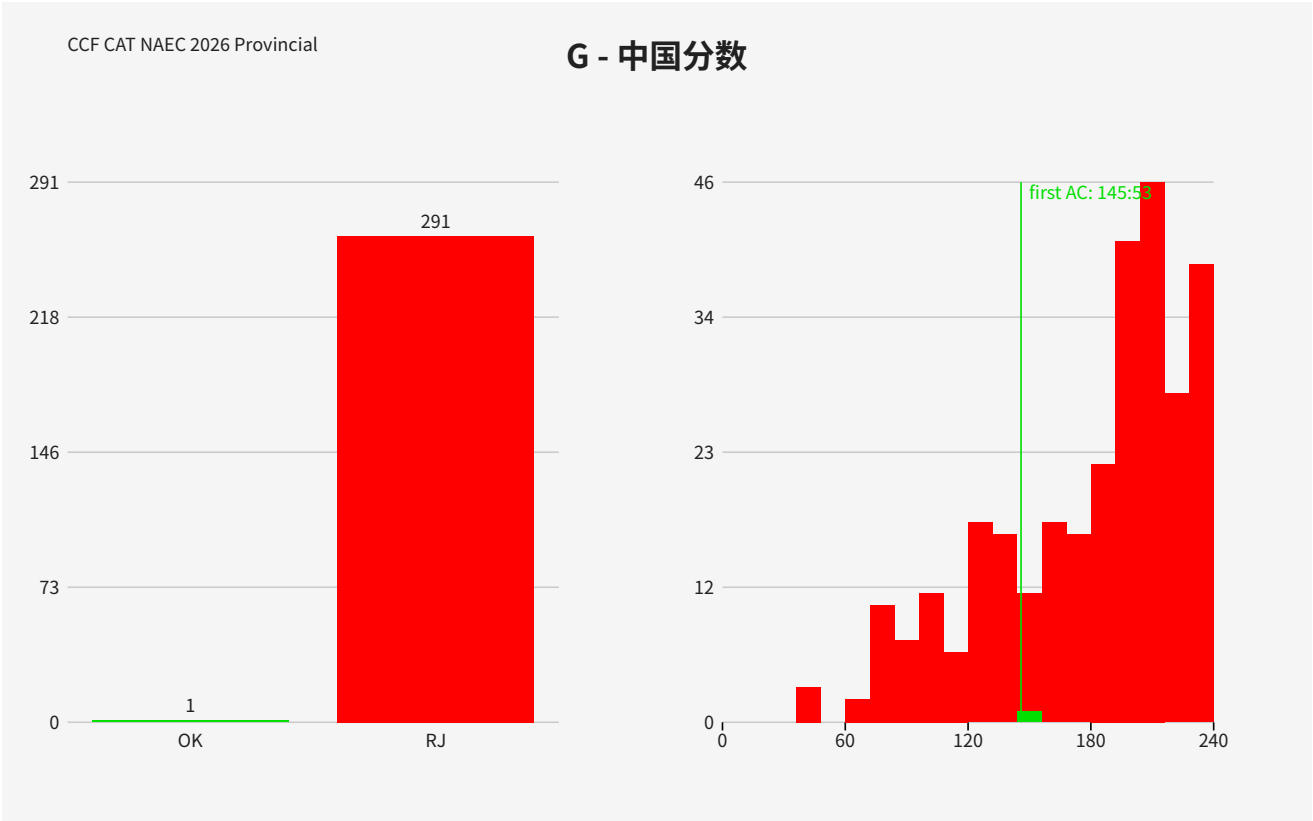
F. 奇怪染料

回到题目本身。

我们要做的事情是，对于每个被 n 自然分割成的彩色块，计算它的颜色段的个数 L ，并将 $\lceil \frac{L+1}{2} \rceil$ 累加进答案即可。

时间复杂度 $O(n)$ 。

G. 中国分数



G. 中国分数

分类讨论：

- 当 $n = 1$ 时，必定有解。直接输出 k 即可。
- 当且仅当 $n = 2$ 时，可能存在无解的情况。由于 k 的总和不超过 10^6 ，暴力求解即可。
- 当 $n \geq 3$ 时，可以证明一定是有解的。

G. 中国分数

- 首先考虑 $k = 1$ 的情况。
 - 考虑放缩： $1 = 1 - \frac{1}{2} + \frac{1}{2} - \frac{1}{3} + \frac{1}{3} \cdots - \frac{1}{n} + \frac{1}{n}$ ，等号右边共 $2n - 1$ 项。
 - 将放缩的第 $2i - 1$ 项与第 $2i$ 项合并，第 $2n - 1$ 项单独留出。
 - 得到式子： $1 = \frac{1}{2} + \frac{1}{6} + \frac{1}{12} \cdots + \frac{1}{(n-1) \cdot n} + \frac{1}{n}$ ，等号右边共 n 项。
 - 分母中的元素全部小于 n^2 。若不存在正整数 $i \leq n$ 使 $i \cdot (i + 1) = n$ ，该构造经排序后可以完全满足题中条件。

G. 中国分数

- 考虑存在正整数 $i \leq n$ ，使 $i \cdot (i + 1) = n$ 的情况。
 - 原式存在且仅存在一个符合该条件的正整数 i ,
 - 整理后的式子形如： $1 = \frac{1}{2} + \cdots \frac{1}{i \cdot (i+1)} \cdots + \frac{1}{(n-1) \cdot n} + \frac{1}{n}$
 - 放缩式形如： $1 = 1 - \frac{1}{2} \cdots + \frac{1}{i} - \frac{1}{i+1} \cdots + \frac{1}{n-1} - \frac{1}{n} + \frac{1}{n}$
 - 等号两边同时减去 $\frac{1}{i} - \frac{1}{i+1}$ 与 $-\frac{1}{n} + \frac{1}{n}$
 - 整理后的放缩式变为： $1 - \frac{1}{n} = \frac{1}{2} + \cdots \frac{1}{(i-1) \cdot i} + \frac{1}{(i+1) \cdot (i+2)} \cdots + \frac{1}{(n-1)}$ ，等号右边总共 $n - 2$ 项。
 - 此时需将 $\frac{1}{n}$ ，即 $\frac{1}{i \cdot (i+1)}$ 拆分为两项后移至等号右侧。

G. 中国分数

拆分方法很多，下面是其中的一种拆分方法。

- 考虑埃及分数中 $\frac{1}{i}$ 型分数的级数分解式。
- 使用其两项的形式，即 $\frac{1}{i} = \frac{1}{i+1} + \frac{1}{i \cdot (i+1)}$
- 将等式两边同时乘 $\frac{1}{i+1}$ ，等式变为 $\frac{1}{i \cdot (i+1)} = \frac{1}{(i+1)^2} + \frac{1}{i \cdot (i+1)^2}$ ，等式变为所需的拆分形式。
- 分母的大小为 $O(n\sqrt{n})$ ，小于等于 n^2 ，满足条件。

G. 中国分数

- 可以证明，该拆分不会与已有的元素重复。
- 对于 $n \leq 10^6$ 的数量级，可以使用程序在范围内进行简单验证。严格证明见下文。
- 由于 $k = 1$ 的拆分中，所有元素都小于 n^2 ，对于 $k > 1$ 的情况，将所有元素同时乘 k 即可。
- 随后进行排序即可满足元素递增的条件。
- 时间复杂度 $O(n \log n)$ 。

G. 中国分数

下面给出该拆分不会与已有的元素重复的严格证明。

- 该证明可以转化为：证明以下两个等式在 x, i 均为正整数时不成立。
 - 等式 1: $x \cdot (x + 1) = (i + 1)^2$
 - 等式 2: $x \cdot (x + 1) = i \cdot (i + 1)^2$

G. 中国分数

- 假设等式 1 成立，即 x 与 $x + 1$ 互质，且其乘积为完全平方数。

引理 1

- 互质数的性质：若两个互质的正整数之积为完全平方数，则每个因子本身必为完全平方数。
- 因此，存在正整数 a, b ，使得 $x = a^2$ ， $x + 1 = b^2$ 。
- 于是， $b^2 - a^2 = 1$ ，即 $(b - a)(b + a) = 1$ 。
- 由于 $b > a \geq 1$ ，则 $b - a$ 与 $b + a$ 均为正整数且乘积为 1，故必有 $b - a = 1$ ， $b + a = 1$ 。
- 解得 $a = 0$ ， $b = 1$ ，则 $x = 0$ ，与 x 为正整数矛盾。因此，等式 1 不成立。

G. 中国分数

- 令 $p = i + 1$, 原等式 2 变为 $x \cdot (x + 1) = p^2 \cdot (p - 1)$, 其中 $x \geq 1, p \geq 2$ 。
- 由于 x 与 $x + 1$ 互质, p^2 与 $p - 1$ 互质, 若等式 2 成立, 则有 $p^2 \mid x$ 或 $p^2 \mid (x + 1)$ 。
- 对于 $p^2 \mid x$ 的情况:
 - 令 $x = p^2 \cdot b$, $x + 1 = a$, 其中 a, b 均为正整数,
 - 由原方程得: $p^2 \cdot b \cdot a = p^2 \cdot (p - 1) \Rightarrow a \cdot b = p - 1$
 - 又 $x + 1 - x = 1$, 有 $a - p^2 \cdot b = 1$, 即 $a = p^2 \cdot b + 1$
 - 代入得: $(p^2 \cdot b + 1) \cdot b = p - 1 \Rightarrow p^2 \cdot b^2 + b + 1 = p$
 - 由于 $b \geq 1$, 有 $p^2 \cdot b^2 + b + 1 \geq p^2 + 2 > p$, 矛盾。

G. 中国分数

- 对于 $p^2 \mid (x + 1)$ 的情况：
 - 令 $x + 1 = p^2 \cdot a$, $x = b$, 其中 a, b 均为正整数,
 - 由原方程得: $p^2 \cdot a \cdot b = p^2 \cdot (p - 1) \Rightarrow a \cdot b = p - 1$
 - 又 $x + 1 - x = 1$, 有 $p^2 \cdot a - b = 1$, 即 $b = p^2 \cdot a - 1$
 - 代入得: $(p^2 \cdot a - 1) \cdot a = p - 1 \Rightarrow p^2 \cdot a^2 - a + 1 = p$
 - 若 $a = 1$, 则 $p^2 \cdot a^2 - a + 1 = p^2 = p$, 解得 $p = 0$ 或 $p = 1$, 与 $p \geq 2$ 矛盾。
 - 若 $a \geq 2$, 则 $p^2 \cdot a^2 - a + 1 \geq 4p^2 - 1 > p$, 矛盾。
- 两种情况均不成立, 故等式 2 不成立。
- 两个等式均不成立, 故该拆分不会与已有的元素重复。证毕。

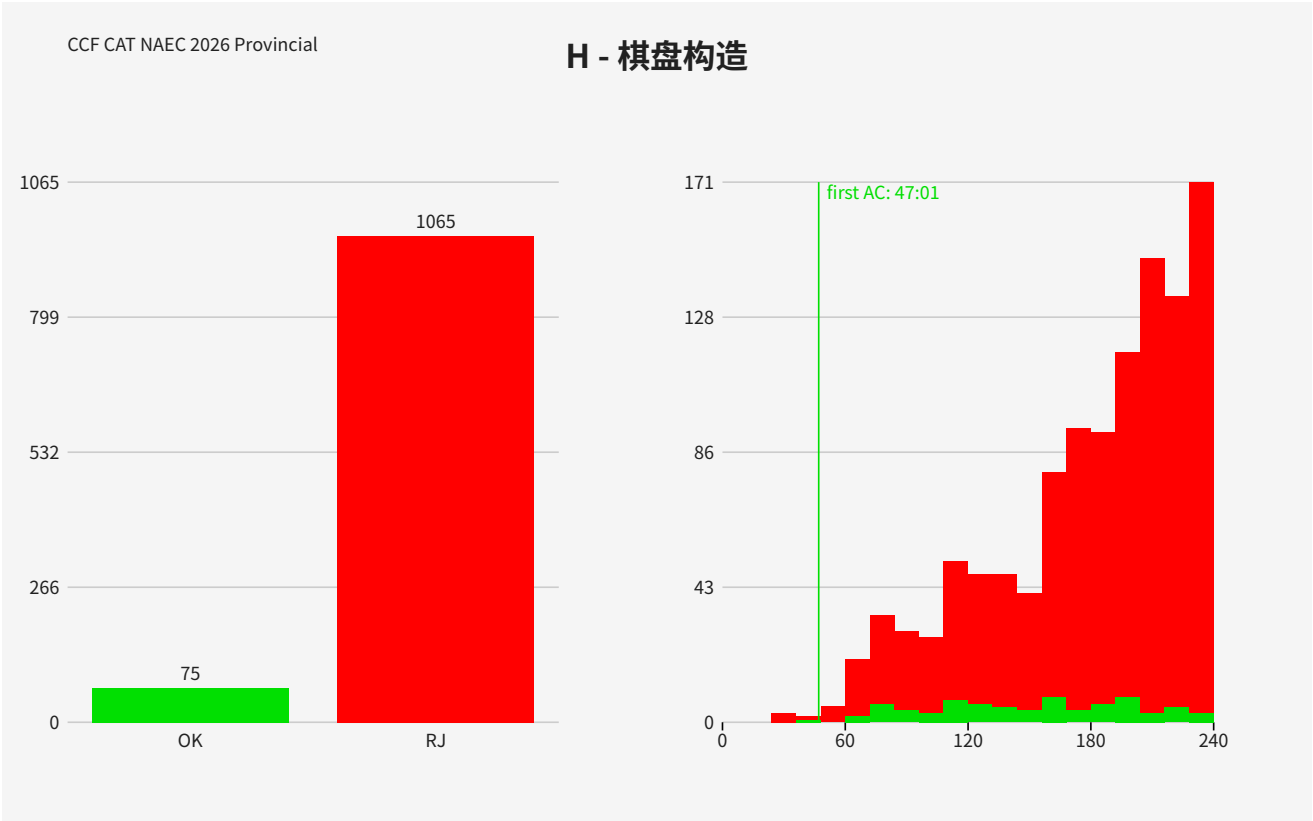
H. 棋盘构造

Prepared by: Rosmarinus

题意

- 在一个 $n \times m$ ($n \times m \leq 10^6$) 的棋盘上放置棋子，每格最多放一个
- 给定每行应放的棋子数 a_i 和每列应放的棋子数 b_i
- 判断是否有解，并给出构造

H. 棋盘构造



H. 棋盘构造

贪心的方法如下：

- 假设我们按照从上到下的顺序逐行放置棋子，并用 b_j 动态维护第 j 列当前剩余的放置需求。每当我们在第 i 行第 j 列放置一颗棋子时，就令对应的 b_j 减一。
- 对于当前第 i 行，设仍有剩余需求的列数为 t （即 $b_j > 0$ 的列数）。
 - 若 $a_i > t$ ，由于每列每行最多放一个棋子，则必然无解。
 - 若 $a_i \leq t$ ，则选择当前 b_j 最大的 a_i 个列放置棋子。
- 重复此过程直至所有行处理完毕。

H. 棋盘构造

接下来给出详细证明：

- 考虑对棋盘第一行的放置方案, 若存在某个合法方案在第一行选择了列集合 H ($|H| = a_1$), 我们需要证明, 若将其替换为贪心选择的集合 G ($|G| = a_1$, 且其中的元素为列需求最多的 a_i 列的编号) 后, 仍可以拓展为合法方案
 - 若 $G = H$, 则无需操作。
 - 否则, 必存在一组 $x \in H \setminus G$ 与 $y \in G \setminus H$ 。且由定义可知, $b_y \geq b_x$ 。

H. 棋盘构造

接下来的目标就是要消除 G 与 H 之间的差异：

- 由于正确方案在第一行选择了第 x 列而没有选择第 y 列。因此在剩下的 $n - 1$ 行中，第 x 列还需放置 $b_x - 1$ 个棋子，第 y 列还需放置 b_y 个棋子。
- 由于 $b_y \geq b_x$ ，因此 $b_y > b_x - 1$ 。即必可以找到第 $i > 1$ 行，使用了第 y 列而没有选择第 x 列。
- 此时，只需让第一行选择 y 而非 x ，让第 i 行选择 x 而非 y 。可以发现，交换前后各行、各列的棋子数均没有发生变化，因此仍为合法方案。
- 重复上述过程，即可以消除 H 与 G 的差异。每次交换后方案仍合法，故使用 G 也可扩展为合法方案。

H. 棋盘构造

在处理完第一行后，问题即可转化为 $n - 1$ 行的子问题。

按照同样的方法进行操作，依据数学归纳法可以证得，若存在一个正确的放置方案，必然可以将其转化为贪心的放置方案。

因此，在存在正确放置方案的前提下，只需要按照贪心的方法放置，必可以得到正确的放置结果。

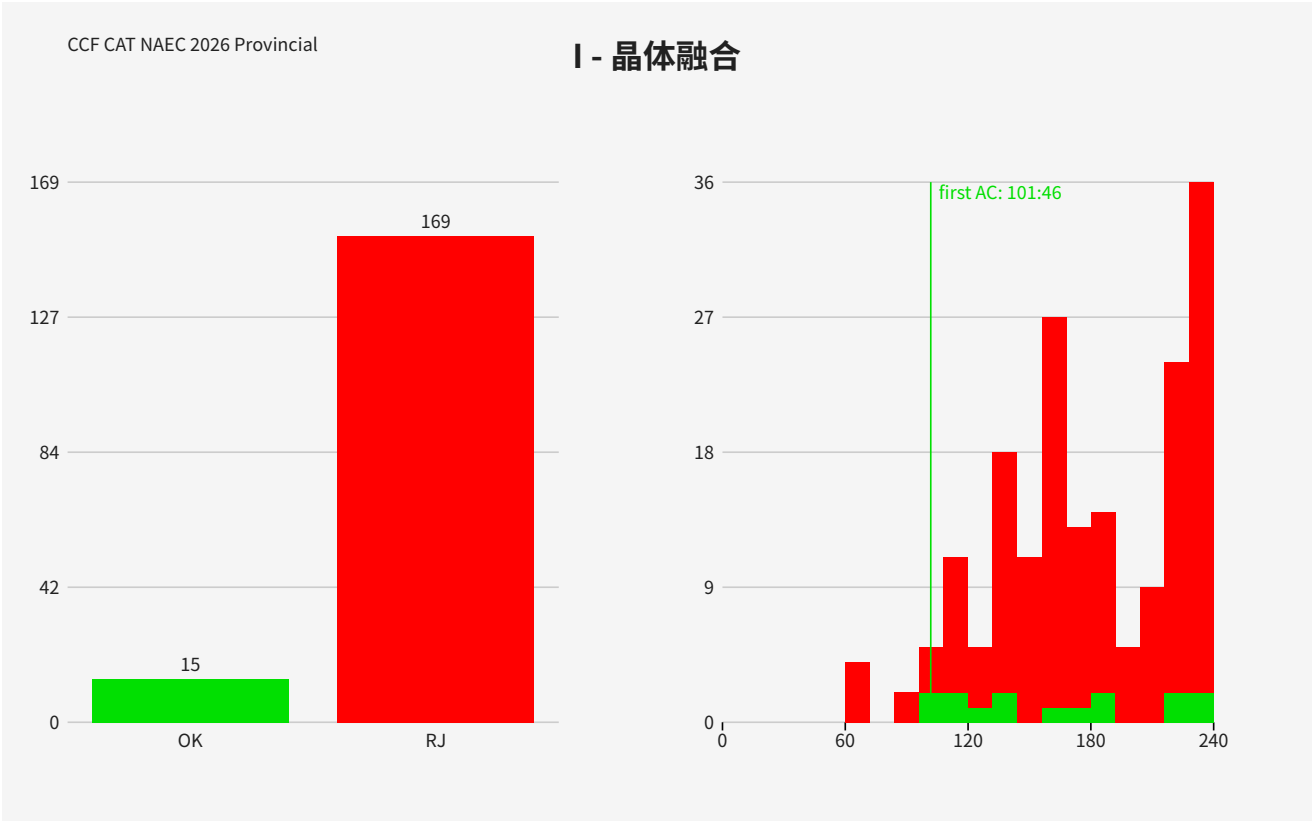
I. 晶体融合

Prepared by: alwaysBelieve

题意

- 有 k 种物品，第 i 种物品大小为 v_i ，数量为 c_i ；
- 每次操作，可以将两个大小为 x 和 y 的物品合并，得到一个大小为 $x + y$ 的物品，并消耗 $f(x, y) = \max\{e \in \mathbb{Z}_{\geq 0} : 3^e \mid \binom{x+y}{x}\}$ 的代价；
- 要求把这些物品合成一个，问代价的总和最小为多少；
- $1 \leq k \leq 10^5, \sum v_i c_i \leq 10^{18}$ ；

I. 晶体融合



I. 晶体融合

- 考虑合并 x 和 y 的代价 $f(x, y) = \max\{e \in \mathbb{Z}_{\geq 0} : 3^e \mid \binom{x+y}{x}\}$ ，它实际就是 $\binom{x+y}{x}$ 中，质因子 3 出现的次数；
- 令 $v_3(n)$ 表示整数 n 中质因子 3 出现的次数。
- 由于 $\binom{x+y}{x} = \frac{(x+y)!}{x!y!}$ 、 $(x+y)!$ 、 $x!$ 、 $y!$ 均为整数，可以知道 $f(x, y) = v_3((x+y)!) - v_3(x!) - v_3(y!)$ ；
- 根据 Legendre 公式：

$$\begin{aligned}
 f(x, y) &= v_3((x+y)!) - v_3(x!) - v_3(y!) \\
 &= \sum_{1 \leq i} \left\lfloor \frac{x+y}{3^i} \right\rfloor - \sum_{1 \leq i} \left\lfloor \frac{x}{3^i} \right\rfloor - \sum_{1 \leq i} \left\lfloor \frac{y}{3^i} \right\rfloor
 \end{aligned}$$

I. 晶体融合

- 令 $\phi(n) = \sum_{1 \leq i} \lfloor \frac{n}{3^i} \rfloor$ ，原问题等价于：
 - 初始时有 m 个物品，第 i 个物品大小为 a_i ；
 - 初始时系统总势能为 $\Phi = \sum_{1 \leq i \leq m} \phi(a_i)$ ；
 - 每次操作，可以将两个大小为 x 和 y 的物品合并，对整个系统贡献 $\Delta(\Phi) = \phi(x + y) - \phi(x) - \phi(y)$ 的势能变化，并产生一个大小为 $x + y$ 的物品；
 - 要求合理安排操作顺序，使得产生的势能的变化的总和尽可能小；

I. 晶体融合

- 利用裂项相消法可以证明，无论合并顺序是怎么样的，最终得到的势能 Φ' 一定等于 $\phi\left(\sum_{1\leq i\leq m} a_i\right)$ ；
- 因此势能的变化总和一定为：

$$\phi\left(\sum_{1\leq i\leq m} a_i\right) - \sum_{1\leq i\leq m} \phi(a_i)$$

I. 晶体融合

- 回到原问题，所求答案即为：

$$\phi\left(\sum_{1\leq i\leq k} v_i c_i\right) - \sum_{1\leq i\leq k} c_i \phi(v_i)$$

- 直接计算即可。
- 其中 $\phi(n) = \sum_{1\leq i} \lfloor \frac{n}{3^i} \rfloor$ 可以在 $O(\log V)$ （其中 $V = 10^{18}$ 表示值域）的复杂度内计算。
- 因此总体时间复杂度 $O(k \log V)$ ；

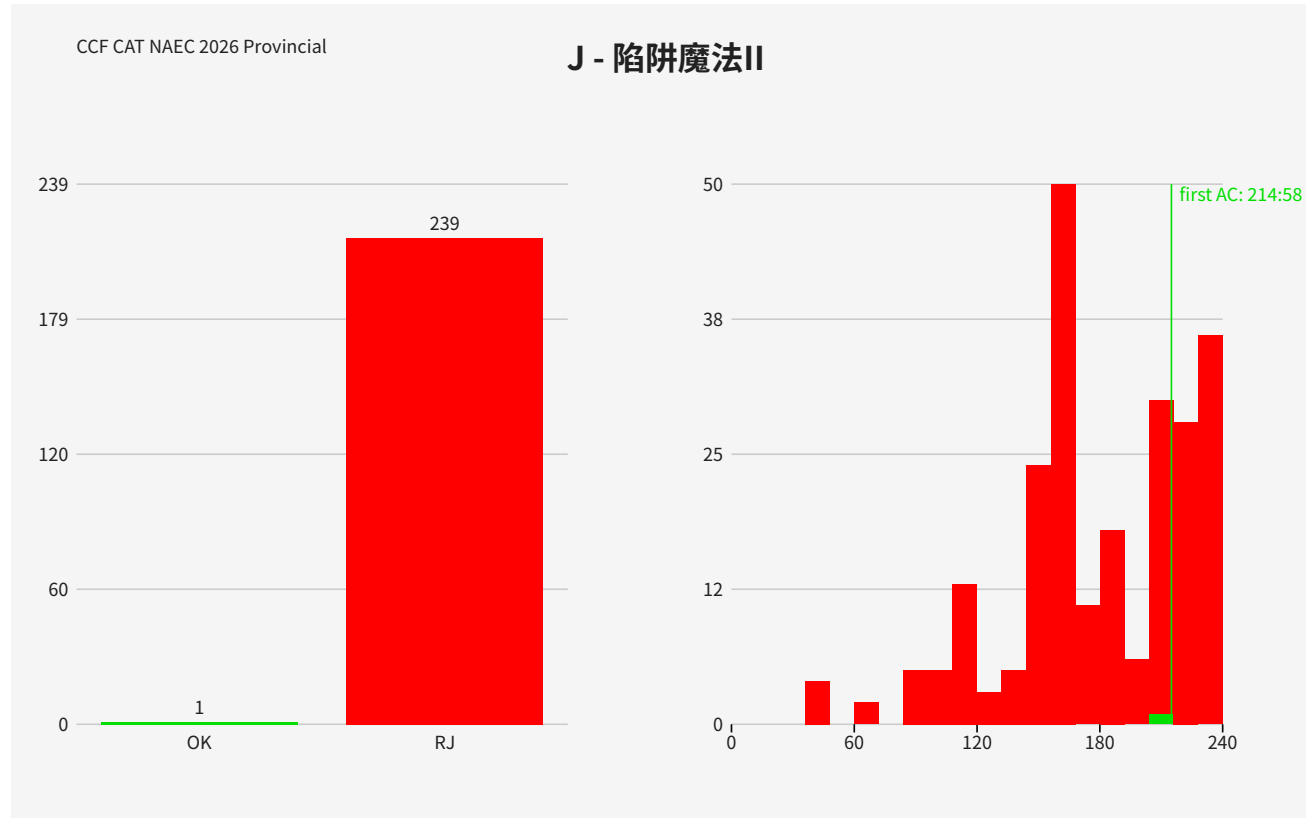
J. 陷阱魔法 II

Prepared by: alwaysBelieve

题意

- 给定一个长为 n 的排列 $[a_1, a_2, \dots, a_n]$ 和一个非负整数 B ;
- 你需要从排列中移除总和不超过 B 的数, 使得 $\max\{a_i - a_j \mid i, j \in \{1, 2, \dots, n\} \wedge i \leq j\}$ 尽可能小;
- 多测, $\sum n \leq 10^5, 0 \leq B \leq \frac{n(n+1)}{2}$;

J. 陷阱魔法 II



J. 陷阱魔法 II

- 随着答案增大，限制只会变弱，因此需要移除的数的总和不会增加，考虑二分答案；
- 令 D 表示被移除的位置的集合，每次 check 实际要求出： $\text{pre}(n) = \max\{a_i - a_j \mid i, j \in \{1, 2, \dots, n\} \wedge i \leq j \wedge i \notin D \wedge j \notin D\}$ 不能超过 m 的情况下，被移除的数字总和 $\sum_{i \in D} a_i$ 至少是多少；

J. 陷阱魔法 II

- 令 $f(x, y)$ 表示，考虑前 x 个数，移除了若干个位置组成集合 D 使得 $\text{pre}(x) = \max\{a_i - a_j \mid i, j \in \{1, 2, \dots, x\} \wedge i \leq j \wedge i \notin D \wedge j \notin D\}$ 不超过 m ，并且保留的数的最大值为 y 的情况下，被移除的数字总和 $\sum_{i \in D} a_i$ 最小是多少；
- 对于空序列，我们认为它的最大值是 0；
- 初始条件下，我们令 $f(0, y)$ ($y \neq 0$) 为 $+\infty$ ， $f(0, 0)$ 为 0；
- 接下来考虑如何从 x 转移到 $x + 1$ ：
- 为了避免混淆，我们令新的保留的数的最大值是 y' ；

J. 陷阱魔法 II

对于 $y' > a_{x+1} + m$:

- 如果保留 a_{x+1} , 会使得 $\text{pre}(x + 1) = y' - a_{x+1} > m$, 不符合要求, 因此必须删去 a_{x+1} 。
- 由于没有新增数字, 因此保留的数的最大值不会改变, 即 $f(x + 1, y')$ 一定由 $f(x, y')$ 转移而来;
- 得到: $f(x + 1, y') \leftarrow f(x, y') + a_{x+1}$, 对于 $y' > a_{x+1} + m$;

J. 陷阱魔法 II

对于 $a_{x+1} < y' \leq a_{x+1} + m$:

- 只要 $\text{pre}(x)$ 没有超过 m , 新放入 a_{x+1} 后, $\text{pre}(x+1) = \max\{y' - a_{x+1}, \text{pre}(x)\}$ 也不会超过 m , 因此最优方案是保留 a_{x+1} ;
- 放入 a_{x+1} 前后, 保留的数的最大值不会发生改变, 即 $f(x+1, y')$ 一定由 $f(x, y')$ 转移而来;
- 得到: $f(x+1, y') \leftarrow f(x, y')$, 对于 $a_{x+1} < y' \leq a_{x+1} + m$;

J. 陷阱魔法 II

对于 $y' < a_{x+1}$:

- 如果保留 a_{x+1} , 则一定会使得保留的数的最大值不再是 y' (因为 a_{x+1} 比 y' 更大), 因此必须删去 a_{x+1} 。
- 由于没有新增数字, 因此保留的数的最大值不会改变, 即 $f(x + 1, y')$ 一定由 $f(x, y')$ 转移而来;
- 得到: $f(x + 1, y') \leftarrow f(x, y') + a_{x+1}$, 对于 $y' < a_{x+1}$;

J. 陷阱魔法 II

最后考虑 $y' = a_{x+1}$ ：

- 由于原序列是一个排列，因此要使得保留的数的最大值是 a_{x+1} 这个数字， a_{x+1} 一定要被保留（因为没有其他数字跟它相等）；
- 如果此前的保留的数的最大值 y 满足 $y > a_{x+1}$ ，放入 a_{x+1} 后，保留的数的最大值一定不是 a_{x+1} （因为 y 比 a_{x+1} 更大），不符合要求；
- 如果此前的保留的数的最大值 y 满足 $y \leq a_{x+1}$ ，放入 a_{x+1} 后，保留的数的最大值一定恰好是 a_{x+1} （因为 y 没有 a_{x+1} 大），并且 $\text{pre}(x+1) = \max\{\text{pre}(x), y - a_{x+1}\} = \text{pre}(x)$ 依旧符合 $\leq m$ 的要求；
- 得到： $f(x+1, y') \leftarrow \min_{y \leq y'} f(x, y)$ ，对于 $y' = a_{x+1}$ ；

J. 陷阱魔法 II

- 取 $\min_{0 \leq i \leq n} f(n, i)$ 即可得到答案。
- 这个 dp 可以使用一个支持「单点修改」「区间加」「查询区间最小值」的线段树维护。
- 因此单次 check 可以做到 $O(n \log n)$ 的复杂度；
- -1 一定属于不可行范围， n 一定属于可行范围，剩下 $O(n)$ 个数需要通过二分检查是否可行，因此需要 check $O(\log n)$ 次；
- 总体时间复杂度 $O(n \log^2 n)$ ；