The 3rd Universal Cup: Hangzhou

Qingyu
SUA 程序设计竞赛命题组
Universal Cup

2024 年 11 月 10 日

概况

- 预期难度：
  - Very Easy： A、K；
  - Easy： E、F、H、M；
  - Medium-Easy： B、G、I、J；
  - Medium： D、L；
  - Medium-Hard： C；
  - Hard： 无。

A. AUS

题意

- 给定三个字符串 S_1, S_2, S_3，判断是否存在一种字符映射方式 $\mathcal{F}$ 使得 $\mathcal{F}(S_1) = \mathcal{F}(S_2) \neq \mathcal{F}(S_3)$。
- $1 \le |S_1|, |S_2|, |S_3| \le 10^3, \sum_{i=1}^{n} |S_1| + |S_2| + |S_3| \le 3 \times 10^4$。

首先处理掉两种 trivial 的情况，*(1)* 如果 $|S_1| \neq |S_2|$，答案一定是 `NO`; *(2)* $|S_1| = |S_2| \neq |S_3|$，答案一定是 `YES`，因为我们可以直接把所有字符都映射到 a。

A. AUS

接下来我们只需要考虑三个串长度相等的情况，不妨设长度为 n。对于每一个位置 i $(1 \le i \le n)$，注意到，如果 $S_1[i] \neq S_2[i]$，那么映射函数中必须要有 $f(S_1[i]) = f(S_2[i])$。另一方面，如果所有这样的限制都满足，一定有 $\mathcal{F}(S_1) = \mathcal{F}(S_2)$。

我们不妨对字符之间在映射函数下的关系建一张无向图，对于所有 i $(1 \le i \le n)$，我们在 $S_1[i]$ 和 $S_2[i]$ 之间连一条边。这张图的每一个连通块就表示了这些字符在映射函数下要相等，那么如果对于所有的位置 i，$S_1[i]$ 和 $S_3[i]$ 都位于同一个连通块，就说明所有满足 $\mathcal{F}(S_1) = \mathcal{F}(S_2)$ 的映射都会让 $\mathcal{F}(S_1) = \mathcal{F}(S_3)$，此时答案就是 `NO`。反之，只需要让每个连通块映射成不同的字符就是一组可行解。

单组数据的时间复杂度是 $O(|S_1| + |S_2| + |S_3|)$。

K. Kind of Bingo

题意

- 有一个 $n \times m$ 的网格以及一个长度为 $n \times m$ 的操作序列，每次操作标记一个格子，每个操作互不相同。按顺序进行操作，如果第 b 次操作后至少有一行都被标记了，且 b 尽量小，那么 b 就是 bingo integer。
- 现在可以交换任意两个操作至多 k 次，最小化 b。
- $1 \le n \times m \le 10^5$，$0 \le k \le 10^9$。

- 枚举最先被标记的是哪一行，设 $i_1, i_2, \cdots, i_m$ 是关于这行的所有操作的下标。如果不进行任何交换，则答案就是 $\min(i_m)$。
- 现在可以进行至多 k 次交换，显然应该把最大的 k 个下标换到前面去，因此答案就是 $\min(i_{m-k})$。当然，答案至少为 m。复杂度 $\mathcal{O}(nm)$。

E. Elevator II

题意

- 有 n 个人正在等电梯，第 i 个人要从 l_i 层到 r_i 层（$l_i < r_i$）。电梯同时只能运送一个人。
- 电梯往上走一层消耗 1 的电量，往下走不消耗电量。电梯一开始在 f 层，求按怎样的顺序运送每个人，才能最小化总电量消耗。
- $1 \le n \le 10^5$，$1 \le l_i, r_i, f \le 10^9$。

- 首先探求答案的下界。
- 因为每个人都要往上走，所以运送每个人的电量（即 $\sum_{i=1}^{n}(r_i - l_i)$）是必须消耗的。
- 另外，由于电梯肯定要到达 $\max(r_i)$。所以还要额外消耗从 f 层到 $\max(r_i)$ 层这一段，没有乘客的区间的电量。

E. Elevator II

- 接下来构造能取到该下界的答案。
- 首先我们要到达 $\max(r_i)$，而且为了不浪费电量，我们要尽可能用乘客的区间覆盖这段路程。因此每次选择起点小于等于当前楼层，但终点大于当前楼层的乘客运送。如果不存在这样的乘客，说明接下来一段没有乘客区间覆盖，找到起点大于当前楼层且最小的乘客运送。这个过程可以使用单调指针 + 优先队列维护。
- 到达 $\max(r_i)$ 之后，剩余乘客按终点从大到小排序运送，这样不消耗任何额外的电量。
- 复杂度 $\mathcal{O}(n \log n)$。

题意

- 给定若干条链，问能否构造出一棵树使得这棵树某一种轻重链剖分的结果恰好是给定的链集合。

首先只有一条链时，直接输出。
下面是 IMPOSSIBLE 的情况：

- 链的长度全部都相等。
- 链的长度最小值和最大值相差 1 ，且有至少两条是最长链。

证明：假设链的长度最大值为 l。

- 对于第一种情况，考虑包含根 u_1 的那条重链 $u_1 \to u_2 \to \cdots \to u_l$。那么可以从 u_l 倒过来交替递归地证明 u_i 的其它子树大小不会超过 $l-i$，进而证明 u_i 不会有其它子树（因为重链长度都为 l）。
- 对于第二种情况，如果包含根的重链长度为 $l-1$，证明思路与第一种情况类似。那现在考虑包含根 u_1 的那条重链 $u_1 \to u_2 \to \cdots \to u_l$，同理，$u_2$ 到 u_l 没有其它子树，u_1 的其它子树都只能是长度为 $l-1$ 的链，不符合有至少两条长度 l 的重链的限制。

H. Heavy-light Decomposition

非 IMPOSSIBLE 情况的构造方法:

- 假如最长链只有一条，那么从根 *root* 出发引一条最长链，其余链全部以 *root* 为父亲
- 最长链大于等于两条，这时最短链的长度 $\leq l-2$，选取一条最长链 $u_1 \to u_2 \to \cdots \to u_l$，最短链以 u_2 为父亲，其它链以 u_1 为父亲

M. Make It Divisible

题意

- 给定序列 a_i。称子数组 $a[l:r]$ 是好的，若子数组里存在一个数可以整除子数组里的所有数。称整个序列是好的，若每个子数组都是好的。
- 给定 k，求满足 $1 \le x \le k$，且使得序列 $a_i + x$ 是好的所有 x 之和。
- $1 \le n \le 5 \times 10^4$，$1 \le a_i \le 10^9$。

- 好的子数组满足：子数组里的最小值可以整除子数组里所有数，即子数组的 gcd 等于子数组的最小值。
- 考虑这样一棵树（笛卡尔树）：将子数组看成节点，子数组去掉最小值后会分成很多区间，这些区间就是下一层节点。整个序列就是根节点。
- 由于所有值都加 x 不会改变元素之间的大小关系，因此无论怎样的 x，笛卡尔树的结构都不会改变。

M. Make It Divisible

- 设子数组 $a[l:r]$ 的最小值为 $a_m + x$。因为子数组的 gcd 等于 $\gcd(|a_l - a_{l+1}|, |a_{l+1} - a_{l+2}|, \cdots, |a_{r-1} - a_r|, a_m + x) = a_m + x$，说明 $a_m + x$ 必须是 $\gcd(|a_l - a_{l+1}|, |a_{l+1} - a_{l+2}|, \cdots, |a_{r-1} - a_r|)$ 的因数。
- 但是对笛卡尔树的每个节点都求 gcd，再分解因数，复杂度可能会达到 $\mathcal{O}(n\sqrt{a_i})$。
- 因此我们只对最上层的节点（整个序列）做一次因数分解，再枚举每个 x，检查每个节点的整除性即可。如果每个节点用 rmq 求 gcd 以及最小值，复杂度 $\mathcal{O}(T\sqrt{a_i} + n(\log n + \log a_i) + nf)$，其中 $f = 1344$ 是 10^9 以内的数最多可能有几个因数。注意第二项不是 $n \log n \log a_i$，因为求 gcd 的复杂度会平摊到 rmq 的每一层。

F. Fuzzy Ranking

题意

- 有 k 个排名表 $A_1, A_2, \cdots, A_k$，定义 (i, j) 是好的当且仅当：
  - 在某个排名表中 i 的排名优于 j，或者：
  - 存在 k，使得在某个排名表中 i 的排名优于 k，且 (k, j) 是好的；
- 多次询问，强制在线，询问 $A_k[l \cdots r]$ 有多少个 pair (i, j) 是好的。
- $1 \leq n, k, q \leq 2 \times 10^5$，$n \cdot k \leq 2 \times 10^5$

F. Fuzzy Ranking

- 相当于区间 $A_k[l, r]$ 的点的导出子图的 SCC 大小的 $\binom{s}{2}$ 之和。
- 注意到，每个 SCC 的断点一定形如 i，满足 $\left|\bigcup_{j=1}^{k} A_k[1, i]\right| = i$。
- 因此可以发现，一组询问一定包含若干个完整的 SCC，以及至多两个（左断点一个、右断点一个）不完整的 SCC。
- 求出所有的断点后，使用前缀和处理询问即可，单独计算中间部分的贡献即可。

F. Fuzzy Ranking

- Fun fact：这个强制在线是最后时刻才加的。

B. Barkeley III

题意

- 给定一个长度为 n 的序列，支持区间按位与一个数，单点修改。询问时给定 l, r，求在区间 $[l, r]$ 中去掉恰好一个数后按位与的最大值。
- $1 \le n, q \le 10^6$。

不难发现，如果某一个二进制位 i 恰有一个数为 0，那么选择这个数后能让答案变大 2^i。于是我们只需要找到一个最高的二进制位，使得恰有一个数在区间内这一位为 1，并且把这个数找到。

B. Barkeley III

一个简单的想法是对于每一位在线段树上维护信息，但这样的复杂度会达到 $O((n+q)\log n \log a_i)$，难以接受。

注意到，如果只想知道哪些位在区间内有恰好一个 0，可以直接用一个压缩状态维护。具体来说，记 f_u 表示线段树上 u 节点对应的区间里哪些位恰有一个 0，g_u 表示区间按位与的结果，那么更新有：

$$f_u = (g_{lson} \ \& \ f_{rson}) \mid (g_{rson} \ \& \ f_{lson})$$

接下来只需要考虑找到这一位对应的数，在线段树上二分即可，时间复杂度 $O((n+q)\log n)$。

J. Japanese Bands

题意

- 有 n_1 张红卡和 n_2 张蓝卡，在每张卡片上写 1 到 m 中的一个数字，同时满足 k 个限制。第 i 个限制是一对整数 (a_i, b_i)，表示能找到一对值分别为 a_i 和 b_i 的异色卡片。求方案数（每种颜色的卡片视为一个 multiset）。
- $1 \le n_1, n_2 \le 10^9$，$1 \le m \le 20$，$1 \le k \le m^2$。

J. Japanese Bands

- 设限制条件里共出现 M 种数，这 M 种数可以分成两类：一类在两种卡片里都要出现，一类恰出现在一种卡片里。
- 设 m_1 表示只出现在红卡上的数的二进制 mask，$|m_1|$ 表示这个 mask 的大小。相似地，设 m_2 表示只出现在蓝卡上的数的二进制 mask。那么剩下的 $(M - |m_1| - |m_2|)$ 种数必须在两种卡片上都出现。而未在限制条件里出现的 $(m - M)$ 种数则可出现可不出现。因此答案为

$$\sum_{m_1} \sum_{m_2 \cap m_1 = \emptyset} f_1(m_1, m_2) \times f_2(m_1, m_2)$$

其中

$$f_1(m_1, m_2) = \binom{n_1 - 1 + (m - M)}{|m_1| + (M - |m_1| - |m_2|) - 1 + (m - M)}$$

$$f_2(m_1, m_2) = \binom{n_2 - 1 + (m - M)}{|m_2| + (M - |m_1| - |m_2|) - 1 + (m - M)}$$

J. Japanese Bands

- 化简一下，答案为

$$\sum_{m_1} \sum_{m_2 \cap m_1 = \emptyset} \binom{n_1-1+m-M}{m-|m_2|-1} \times \binom{n_2-1+m-M}{m-|m_1|-1}$$
$$= \sum_{m_1} \binom{n_2-1+m-M}{m-|m_1|-1} \sum_{m_2 \cap m_1 = \emptyset} \binom{n_1-1+m-M}{m-|m_2|-1}$$

- 因此可以用高维前缀和预处理 $\sum_{m_2 \cap m_1 = \emptyset} \binom{n_1-1+m-M}{m-|m_2|-1}$，再枚举 m_1 即可。
- 另外 m_1 里的每对元素不能出现在同一个限制条件里，m_2 同理。因为 m 只有 20，所以可以用一个 int 把和每个数有关的限制条件存下来，枚举 m_1 和 m_2 时通过位运算检验一下合法性即可。复杂度 $\mathcal{O}(m \times 2^m)$。

I. Identify Chord

题意

- 给定一个长度为 n 的环，环上添加了一条你不知道具体位置的弦
- 每次你可以询问两个点 u, v 并获得它们的最短路长度
- 需要用最多 40 次询问获取隐藏的弦的具体位置
- $n \leq 10^9$。

I. Identify Chord

- 先询问 $u=0, v=(n+1)/2$ ，如果答案为 $n/2$，则我们需要旋转 u, v 直到答案小于 $n/2$ 为止。
- 偶数的旋转顺序是 $(1, \frac{n+1}{2}+1), (2, \frac{n+1}{2}+2), \cdots$
- 奇数的旋转顺序是 $(1, \frac{n+1}{2}), (1, \frac{n+1}{2}+1), (2, \frac{n+1}{2}+1), (2, \frac{n+1}{2}+2), \cdots$
- 现在 u 到 v 的最短路一定经过隐藏弦，并且假设由于这条隐藏弦，距离减小了 d 。询问 $(u+1, v)$ 和 $(u-1, v)$ 来确定 u 到 v 的最短路是沿着往哪个方向。
- 沿着这个方向二分，找到最大的 mid 使得 $dis(mid, v)$ 满足减少的距离为 d ，这样 mid 是隐藏弦其中一个端点，通过减少的距离 d 可以 $O(1)$ 次询问找到另外一个端点
- 询问次数为 $1 \times \log(n) + O(1)$。

I. Identify Chord

- 错误的随机方式：
  一直随机距离为 $\frac{n}{2}$ 的询问对 (u, v)，直到它们真正的距离小于 $\frac{n}{2}$ 为止，然后继续第二步及之后的步骤
- 在 n 为奇数时，弦的长度为 2 时，单次随机有 $\frac{1}{2}$ 的概率距离不变，那么有 $\frac{1}{2^{10}}$ 的概率多增加 10 步，这样会超出步数限制
- 不太良好的交互习惯：
- 不查看交互结束之后是否需要读入返回值

G. Gathering Mushrooms

题意

- 给定 n 个点 n 条边的有向图，每个点恰有一条出边，且每个点有个数字 a_i。
- 问从每个点出发，记录经过的节点上出现的数字，最先出现 k 次的数字是什么。
- $1 \leq n \leq 2 \times 10^5, 1 \leq k \leq 10^9$。

G. Gathering Mushrooms

- 一道偏重实现的题目。给定的图是一个有向的环套树。设第 i 个点指向第 p_i 个点，那么第 i 个点的答案要么和第 p_i 个点相同，要么就是 a_i。所以我们要比较两者的步数谁比较小。
- 考虑这样的实现方式：把环从任意地方断开，然后把环延长一倍。这样一个连通块就变成了一棵树，直接按树的 DFS 序求答案即可。因为环延长了一倍，所以计算答案的时候相当于先在环上转了一圈，再进入其它节点，可以求出正确答案。
- DFS 的过程中可以用 vector 简单维护出每种数往树根方向走几步才能出现 k 次。如果走到树根才出现了 t 次，则答案再加上 $\lfloor \frac{k-t}{c} \rfloor \times l + s_{k \bmod t}$，其中 c 是这种数在环上出现了几次，l 是环长，s_i 是这种数在环上第 i 次出现时的步数。复杂度 $\mathcal{O}(n)$。

D. Dividing Sequence

题意

- 给定长为 n 的序列 A，你需要把他分割成两个子序列 B 和 C，使得 B 的字典序不大于 C，且 C 的字典序最小，输出最小的 C。
- $n \le 5\,000, \sum n \le 10^4, |\Sigma| \le 10^5$。

- 这里提供两个做法，第一种做法 $O(n^2)$ ，第二种做法是线性的。

D. Dividing Sequence

做法一

- 考虑逐位枚举答案的前缀，然后 dp
- f_i 表示对原串长为 i 的前缀，序列 C 匹配了 ans，B 匹配了 ans 的前缀，ans 的下一位最小是多少，ans 是你枚举的前缀。
- 每一轮，拿出合法的 f_i 里 f_i 最小的值，拓展一位答案，然后更新所有合法的 f_i。
- 一共 $O(n)$ 轮，每轮状态数 $O(n)$，更新 $O(1)$，总复杂度 $O(n^2)$。

D. Dividing Sequence

做法二

- 考虑对原串做 lyndon 分解出的 m 个串 $S_1, S_2, \ldots, S_m$，我们有 $S_1 \geq S_2 \geq \ldots, \geq S_m$。
- i 从 1 开始，若 $S_{2i-1} = S_{2i}$，则将这两个串分别接在 B, C 的末尾，$i++$
- 若 $S_{2i-1} \neq S_{2i}$，将 S_{2i-1} 接到 C 之后，剩下的所有串按顺序接到 B 的末尾，此时的 C 就是答案。
- 总复杂度 $O(n)$，暴力的 $O(n^2)$ lyndon 分解实现也可以通过。
- 我们证明这样构造的结果是最优的。

引理 0

设 s 是 Lyndon 串。不能将 s 分为两个 $\leq s$ 的子序列，除非其中之一为 s。

证明：

- 设我们成功把 s 分解为 a 和 b 两个子序列，且 a 和 b 都严格小于 s。s 的长度至少为 2。
- 不妨设 s 的首个字符分给了子序列 a。设 s 的首个字符为 $0, s = 0t, a = 0u$。
- 若 u 为空，$b = t$。又因为 s 是 Lyndon 串，t 为 s 的前缀或者 $t > s$。但是 $b < s$。所以 t 必须为 s 的前缀。所以 s 的所有字符为 0，这与 s 是 Lyndon 串矛盾。所以 u 非空。

继续证明：

- 若 u 非空，此时，若 $t > s$，则 $b < t$ 且 $u < t$ ，即 t 可分解为两个严格比它小的非空子序列。
- 因为 s 是长度大于 1 的 Lyndon 串 $0t$，t 不能是 s 的前缀。所以只有唯一一种可能，即我们成功获得了 t，它长度比 s 少 1，且可分解为两个严格比它小的非空子序列。由于 s 的长度是有限的，归纳的证明可得矛盾。
- 于是引理 0 得证。

D. Dividing Sequence

做法二证明

引理 1

设 $s \geq t$ 为两个 Lyndon 串。则任意 Lyndon 分解开头两个 Lyndon 串为 s，t 的串 x 划分给满足题目要求的 b, c 的最优解中，x 开头的 s 必须全部归属 c。

证明：

- 由于存在一种 c 以 s 开头的合法解（前文给出的做法），所以最优解中，x 开头的 s 中归属 c 的部分必须 $\leq s$。由于最优解中 $b \leq c$，x 开头的 s 中属于 b 的部分同样 $\leq s$。
- 首先若 $s > t$ ，我们考虑 s 分给 b, c 的部分，根据引理 0，我们无法做到 b, c 同时 $< s$，因此最优解即为 $c = s$。
- 若 $s = t$，考虑我们将 s 划分给 b, c 的两部分，根据前面的分析，我们需要把 Lyndon 串 s 分为两个 $\leq s$ 的子序列，由引理 0，必有一个子序列为 s。

继续证明:

- 现在我们要证，当 $s = t$，x 开头的 s 全归属于 b，且 c 的开头也等于 s 的情况下，t 归属于 c。
- 此时与前文分析类似地，根据引理 0，仍然是对于 t，我们无法分成两个 s 的。
- 如果分给 b 或者分给 c 的其中一边 $> s$，这时可以发现已经不优于我们给出做法的构造了。
- 因此仍然是整个 t 分给 b 或者 c，如果分给 c 我们证明的东西就达成了，于是只需要考虑仍然分给 b 的情况。这时加了两个 s，于是按这个思路迭代下去，因为 Lyndon 分解不增，可以发现这样不优。

引理 2

设 s 为一个 Lyndon 串。则任意 Lyndon 分解前两个 Lyndon 串均为 s 的字符串 x 的答案中，开头的两个 s 必须分属 c 和 b。

证明：

- 根据引理 1，开头的第一个 s 归属于 c。
- 考察第二个 s，根据引理 0，他也是不能被分成两个 s 的。并且如果分出来一个 $> s$ 的，分给 b 会导致不合法（因为已经将一个 s 分给 c 了），分给 c 不优于我们的构造。所以只能将第二个 s 整个给 b 或者 c。
- 如果给 s 整个分给 b，引理得证。如果整个给 c，我们可以迭代证明这样不优于我们的构造，具体过程略。

根据引理 1,2 可得上文给出的构造是最优的解。

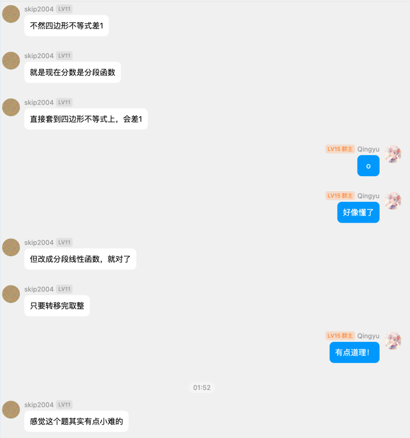
L. Let's Go! New Adventure

题意

- 给定经验值序列 a_i 和从 $i-1$ 级升到 i 级需要的经验 b_i，保证 $b_i \le b_{i+1}$。
- 将 a_i 分段，每段的贡献为经验和能够升到的等级减 c，求最大贡献和。
- $n, m, c \le 5 \times 10^5$，$\sum a_i, \sum b_i \le 10^{12}$

L. Let’s Go! New Adventure

TLDR（一句话省流题解）



L. Let's Go! New Adventure

- 考虑动态规划:

$$dp_j = \max_{0 \le i < j} \{dp_i + w(i, j)\}$$

$$dp_0 = 0$$

- 贡献函数 w 为

$$w(i, j) = w(A_j - A_i) = \max\{k \mid B_k \le A_j - A_i\} - c$$

其中 $A_i = \sum_{j=1}^{i} a_j$, $B_i = \sum_{j=1}^{i} b_j$。

引理 0

对于 $i < j < k < l$，有 $w(i,l) + w(j,k) \le w(i,k) + w(j,l) + 1$。

证明：

- 首先将 B 和 w 定义延展到非负实数域：

$$B_{ext}(x) = B_{\lfloor x \rfloor} + \{x\} \times b_{\lfloor x \rfloor + 1}$$

$$w_{ext}(x) = B_{ext}^{-1}(x) - c$$

- 显然有 $B_i = B_{ext}(i), w(i) = \lfloor w_{ext}(i) \rfloor$。
- 由于 $B_{ext}(x)$ 为不减凸函数，可得 $B_{ext}^{-1}(x)$ 为不减凹函数。
- 对于 $x > y$ 以及 $\Delta > 0$ 有：

$$B_{ext}^{-1}(x+\Delta) - B_{ext}^{-1}(x) \le B_{ext}^{-1}(y+\Delta) - B_{ext}^{-1}(y)$$

继续证明：

- 即

$$w_{ext}(x+\Delta) - w_{ext}(x) + w_{ext}(y) - w_{ext}(y+\Delta) \le 0$$

- 则

$$\begin{aligned}
& w(i,l) - w(i,k) + w(j,k) - w(j,l) \\
&= \lfloor w_{ext}(A_l - A_i) \rfloor - \lfloor w_{ext}(A_k - A_i) \rfloor \\
&+ \lfloor w_{ext}(A_k - A_j) \rfloor - \lfloor w_{ext}(A_l - A_j) \rfloor \\
&< w_{ext}(A_l - A_i) - (w_{ext}(A_k - A_i) - 1) \\
&+ w_{ext}(A_k - A_j) - (w_{ext}(A_l - A_j) - 1) \\
&\le 2
\end{aligned} \tag{1}$$

由于 w 取值始终为整数，得证。

L. Let's Go! New Adventure

- 根据引理可得对于 $i < j < k < l$，有

$$(dp_i + w(i,k)) - (dp_j + w(j,k)) + 1 \geq (dp_i + w(i,l)) - (dp_j + w(j,l))$$

- 那么

$$(dp_i + w(i,k)) < (dp_j + w(j,k)) \rightarrow (dp_i + w(i,l)) \leq (dp_j + w(j,l))$$

即若当前位置 k 上后决策点 j 严格优于前决策点 i，则后继位置 l 上 j 不会劣于 i。

- 得到决策单调性，但这里只保证后决策点严格优的位置一定在严格劣的位置之前，两决策点相等的位置可以任意不连续地出现。

L. Let's Go! New Adventure

- 考虑二分队列求解：DP 时记录目前为止每个决策点的最优区间，新插入的决策点会覆盖之前的一些区间。
- 单点求值需要在 B_i 上二分，单次复杂度 $\mathcal{O}(\log n)$。
- 如何求两决策点对应区间分界点？直接二分无法确定相等的情况属于哪一边。
- 使用 w_{ext} 代替 w 求值，由于 w_{ext} 严格符合四边形不等式，可以正常二分。总复杂度 $\mathcal{O}(m + n \log n \log m)$。
- 或注意到后决策点第一次严格优的位置一定对应某次升级，且升级位置可以正常二分，求得分界点所对应的最小 A_i，总复杂度 $\mathcal{O}(m + n \log m)$。

C. Catch the Star

题意

- 给定凸包 S 代表目标和另外 n 个凸包 M_i 代表障碍物，找到线段 $(l, 0)$ 到 $(r, 0)$ 之间（不包括端点）能完全无遮挡看到目标的点集有多长。空集输出 -1。
- 保证目标、障碍物、给定线段互相没有交集，但障碍物内部互相之间可交。视线切障碍物时视为无遮挡。
- $n \leq 10^4, k_i \leq 10^5, \sum k_i \leq 10^6$。

C. Catch the Star

- 首先在凸包上用循环二分找到每个障碍对目标的两条内切线。
- 假设切线分别为 $\overrightarrow{m_{in}s_{out}}$ 和 $\overrightarrow{m_{out}s_{in}}$，则由 $\overrightarrow{m_{in}s_{out}}$，$\overrightarrow{m_{in}m_{out}}$，$\overrightarrow{s_{in}m_{out}}$ 三个开半平面共同覆盖的区域内目标会被障碍物遮挡。注意特判 m_{in} 与 m_{out} 为同一点的情况。
- 用给定线段与三个半平面分别求交后再求交，得到的线段就是被该障碍物遮挡的部分。线段与半平面求交只需判断端点是否在半平面内，随后根据情况求直线交点即可。
- 注意这里需要使用有理数；由于给定线段在 x 轴上求交点和求交时只需要计算 x 的值。
- 最后将遮挡线段的端点排序，扫描线求合法线段长度。注意原线段与遮挡线段均为开区间。

C. Catch the Star

最后

- 没听明白？没关系。
- 访问 https://sua.ac/wiki/ ，有文字版题解与带注释的参考代码。

Thank you!