2026 年 ICPC 国际大学生程序设计竞赛
全国邀请赛 (深圳)

SUA 程序设计竞赛命题组

2026 年 4 月 11 日

A. Greetings From Prof. Chen

题意

- 给定正整数 n，构造一个仅由小写英文字母组成的字符串，使其 ASCII 码之和恰好等于 n，或指出无解。
- $\sum n \leq 10^7$。

A. Greetings From Prof. Chen

- 假设字符串的长度为 m，那么 $97m \le n \le 122m$ 都有解，只需从长为 m 的全 a 串开始逐字符往上加。
- 任取一个合法的 m 即可，不存在就无解。时间复杂度 $O(n)$。

B. All-Star Showdown

题意

- 给定 n 个带权三维点，每个点红蓝二染色，要求任意两个同色点之间的欧式距离至少是 d，对每个 $k = 1, 2, \ldots, n-1$ 求出有恰好 k 个红点的染色方案中红点的权值之积的和，答案对 $998\,244\,353$ 取模。
- $n \leq 10^5$。

B. All-Star Showdown

- 对 n 个点建图，如果两个点之间的距离 $< d$，连一条边表示它们不能同色，那么这个图是二分图才有合法的染色方案。
- 空间按照 $d/2 \times d/2 \times d/2$ 进行网格划分，同一格子里任意两个点的距离 $< d$，那么每个格子里最多能有 3 个点。
- 对每个点，枚举所在格子附近 $5 \times 5 \times 5$ 格子内的点检查是否要连边。由于对称性，可以只检查其中一半的格子。
- 这部分时间复杂度是 $O(n)$，有一个维度在指数上的常数。

B. All-Star Showdown

- 建图之后判断二分图，如果是二分图就有合法的染色方案。
- 对每个连通块，设两部分点数分别为 a 和 b，点权之积分别为 v_a 和 v_b，可以得到一个多项式 $(v_a x^a + v_b x^b)$。
- 分治 FFT 把所有连通块的多项式乘起来即可，这部分时间复杂度是 $O(n \log^2 n)$。

C. One Item Away

题意

- 现在有 n 个人 m 个物品。对每个人 i 来说，每个物品集的子集 $s \subseteq 2^{[m]}$ 有一个**隐藏**的价值 $u_i(s)$。
- 所有的价值函数 u_i 保证 $u_i(\emptyset) = 0$，集合变大价值不减。
- 每次询问你可以指定一个人 i 和一个子集 s，交互器会告诉你 $u_i(s)$ 的价值。
- 你需要询问不超过 $O(nm)$ 次后，实现符合如下要求的分配：
  - 最终每个物品分给了恰好一个人；令 i 拿到的集合为 s_i。
  - 对于任何两个人 i, j $(1 \le i, j \le n)$：
    如果 $u_i(s_j) > u_i(s_i)$，存在 $x \in s_j$ 满足 $u_i(s_j \setminus x) \le u_i(s_i)$。
- $n \le 100, m \le 500$。

C. One Item Away

- 我们称 $u_i(s_i) < u_i(s_j)$ 为 i **嫉妒** j 。
- 先思考只有两个人的时候怎么做。
- 考虑对物品**逐个**进行分配，过程中**始终保持分配符合要求**。
- 初始两个人的物品集都为空。每次根据当前情况决定：
- (i) 如果两个人**都嫉妒**对方，则交换两人集合变为情况 (ii)。
- (ii) 如果两个人**都不嫉妒**对方，则将当前物品给随意一个人。这样只可能让没得到物品的那个人嫉妒，且符合题目要求。
- (iii) 如果只有一个人嫉妒对方（假设 1 嫉妒 2），则把当前物品给 1。容易证明新的分配仍然符合要求。

C. One Item Away

- 考虑将这一思路扩展到 n 个人。
- 考虑对物品**逐个**进行分配，过程中**始终保持分配符合要求**。
- 建立一张 n 个点的有向图，边 $i \to j$ 表示 i 嫉妒 j 。
- 初始所有人的物品集都为空。每次考察图中是否有环：
- 若图中存在一个有向环，则将环上所有人的集合沿着环的反方向移动一条边。这样环上的边都会被消除，一些旧的边会改动起点/终点，但不会引入新的边。一直操作直到无环。
- 图变为 DAG 后，一定存在至少一个 0 入度点（无人嫉妒）。将当前物品给这个点。考虑分配带来边的变化：
  - 该点对其他点的嫉妒：该点自己的集合价值不减，这些嫉妒依旧符合题目要求，即 $u_i(s_i \cup \{y\}) \geq u_i(s_i) \geq u_i(s_j \setminus \{x\})$ 。
  - 其他点对该点的嫉妒：由于此前该点为 0 入度点，因此所有的嫉妒都是新产生的，这些新产生的嫉妒天然的符合题目要求（删掉新添加的物品即可消除）。

C. One Item Away

- 上述做法只需要在每次分配物品后，询问所有人对这个物品所在集合的价值，恰好使用 nm 次询问。
- 每次分配物品后，新产生的边必定指向获得该物品的点，因此单次至多产生 $n-1$ 条边，总共产生 $O(nm)$ 条边。
- 每次反向旋转一个环至少会消除一条边，因此最多进行 $O(nm)$ 次找环 $+$ 旋转操作。
- 找环使用朴素的 BFS/DFS 即可在 $O(n^2)$ 内实现。旋转同理。
- 这样总复杂度 $O(n^3m)$ 足够通过本题。
- 精细地分析和实现可以得到更低的复杂度。

D. City Management

题意

- 您正在管理一些工厂和工人，每座工厂最多只能有一个工人在里面工作。每座工厂有一个产能值 x_i，表示在一轮生产中，若有一名工人在里面工作，则会产生 x_i 的收益。
- 您已经预知，在将来的 n 天，每天会发生如下两种事件之一：
- 事件 1：有一名工人来找工作。这位工人到来后，您可以重新分配所有工人（包括已经在工厂里工作的工人）的岗位，或让他待岗。
- 事件 2：有一座产能值为 x_i 的工厂被建立。这座工厂被建立后，如果有待岗的工人，您可以选择是否让一名待岗工人去该工厂工作。当然，您也可以让工厂暂时空置。
- 当天的事件执行完毕后，所有工厂都会进行一轮生产，并根据前文所述规则产生收益。您需要求出最大总收益。
- 事件数量 $n \leq 2 \times 10^5$。

D. City Management

- 按 W 操作分段，每一段之间是独立的，因为在进入下一段的时候可以将工人任意地重排。
- 考虑一个长度为 len 的段，设在这个段之前有产能为 $x_1, x_2, \cdots, x_m$ 的工厂建立，在这个段内部依次有产能为 $y_1, y_2, \cdots, y_{len-1}$ 的工厂建立。
- 考察在进入本段之前的那一次 W 操作，您可以把一个工人放到产能为 x_i 的工厂，获得 $x_i \times len$ 的收益，您也可以让一个工人待岗，等产能为 y_j 的工厂建立时，让他去工作，获得 $y_j \times (len - j)$ 的收益。
- 令 W 为工人的数量，我们需要求解一个“前 W 大的和”的问题。考虑使用两个 set S_1, S_2，S_1 维护前 W 大的收益，S_2 维护其余的收益。

D. City Management

- 棘手的一点是，我们需要对 $x_1, x_2, \cdots, x_m$ 进行一个整体 $\times len$ 的操作。为了解决这个问题，我们不直接将收益 $profit$ 丢到 set 里面，而是将一个 pair：$(\lfloor profit/len \rfloor, profit \bmod len)$ 放进去，这样，产能为 x_i 的工厂会始终用一个 $(x_i, 0)$ 的 pair 去刻画。
- 经过上述分析，我们可以用如下方式去统计一整段的答案：

详细步骤

- 将所有的 $y_j \times (len - j)$ 转化成 pair，丢到 set 中。
- 调整 set 的大小，使得 S_1 的大小等于 W 或 S_2 非空。
- 计算答案。
- 将所有 $y_j \times (len - j)$ 对应的 pair 从 set 中删除。
- 然后对于所有 y_j，将 $(y_j, 0)$ 丢到 set 中。

D. City Management

- 至于复杂度，定义势能函数为：S_1 的大小和 W 的差的绝对值。那么，一个新的工厂会让势能 $+O(1)$，同时，每调整一次 set 的大小，都会让势能 -1，所以 set 的操作不超过 $O(n)$ 次，复杂度 $O(n \log n)$。

E. Card Checking

题意

- 此题需要求解一个三人斗地主残局，当地主只剩一张牌时，求农民是否有必胜策略。
- 出牌只允许出单张和对子，且对子不能拆成两个单张。所有人的手牌均对所有人可见。
- 出牌顺序按照 $A \to B \to C \to A \cdots$ 的顺序循环出牌，其中 A, B 是农民，C 是地主。第一轮手牌 A 先出。
- 牌的点数范围 $n \leq 2.5 \times 10^5$，每一种牌每人最多拥有两张。

E. Card Checking

- 我们定义“小牌”为点数小于地主的牌，“大牌”为点数大于等于地主的牌。
- 如果让 B 先出完，是简单的，您需要满足下面两个条件：

条件

- B 的手牌中只有一张小牌。
- A 能把牌传过去。

- 这里有一些边界情况，例如 A 如果用单张传，那么 B 一定要消耗掉一张大牌。但如果 B 手上只有一张牌，那么它可以直接胜利，不一定需要消耗大牌。

E. Card Checking

- 如果让 A 先出完，可以通过相互配合消耗掉 A 手上多余的小牌，如下两种操作都可以消耗掉一张小牌：

操作

- A 出小牌，B 出一张大牌，然后 A 跳过。下一轮中，B 出一个对子，A 出一个更大的对子。
- A 出小牌，B 出一张大牌，然后 A 出一张更大的大牌。

- 先处理第一种操作，我们贪心地求出二人手上的对子分别能支持多少次第一种操作，由于“小的大牌”是在第二种操作中有用的，所以对每一次操作消耗一张“大的大牌”。
- 然后，我们根据 A 中剩余的小牌张数，求出第二种操作需要几次。这个时候，我们考察 B 中最小的几张大牌，和 A 中最大的几张大牌，看是否符合大小关系即可。

F. Astra

题意

- 给定一棵 n 个节点的树，两人轮流进行操作。
- 一开始节点 s 被涂红，之后每次操作可以涂红 1 到 k 个节点，要求涂红的第一个节点是红色节点的邻居，后续节点是上一个节点的邻居。每个节点只能涂一次。
- 把最后一个节点涂红的人获胜，对于每个 $1 \le s \le n$ 问谁赢。
- $n \le 2000$

- 平等博弈用 sg 函数计算。设 $f(u \to v)$ 表示 $u \to v$ 这条边指向的子树的 sg 值。
- 计算时，遍历所有从 v 出发的，长度小于等于 k 的路径，计算每种情况的 mex。一条路径把以 v 为根的子树，分隔成互不影响的若干棵子树，可以把它们的 sg 值异或起来。
- 可以使用记忆化搜索（写成递归的 dp）实现，复杂度 $O(n^2)$。

G. Snake

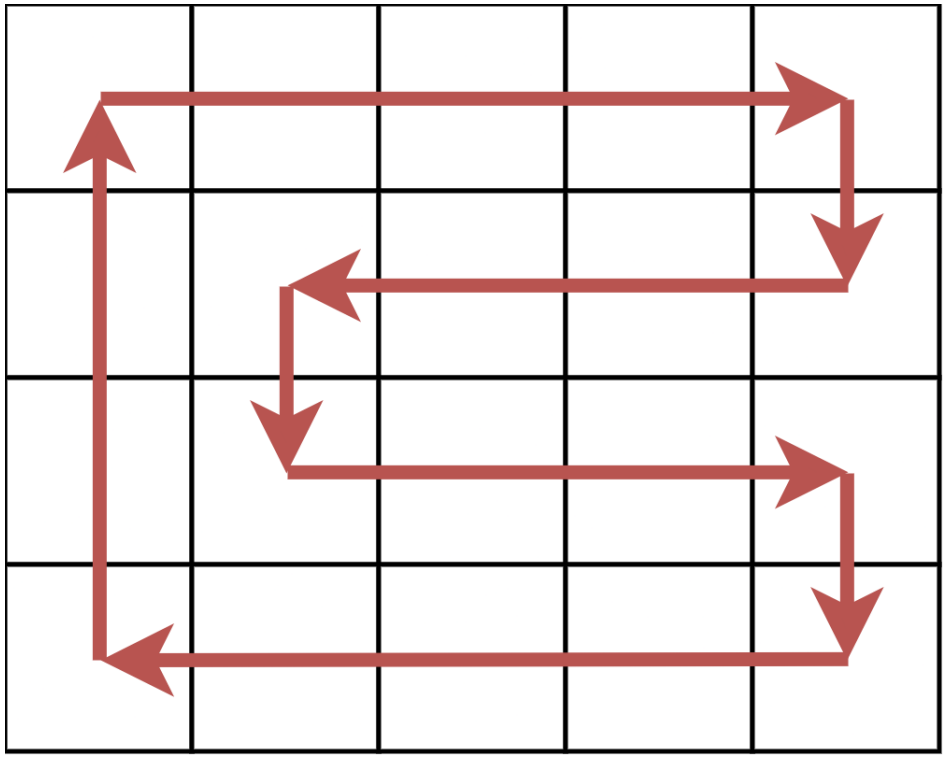
题意

- 交互题。在 $n \times m$ 且没有障碍的网格上玩贪吃蛇，一开始蛇的长度为 1，$(nm-1)$ 个苹果依次出现，需要把它们依次吃掉，最后蛇占满网格。
- $n \times m \leq 100$

- 让蛇在一条环线上一直走，就能吃完所有苹果，而且不会碰到自己。

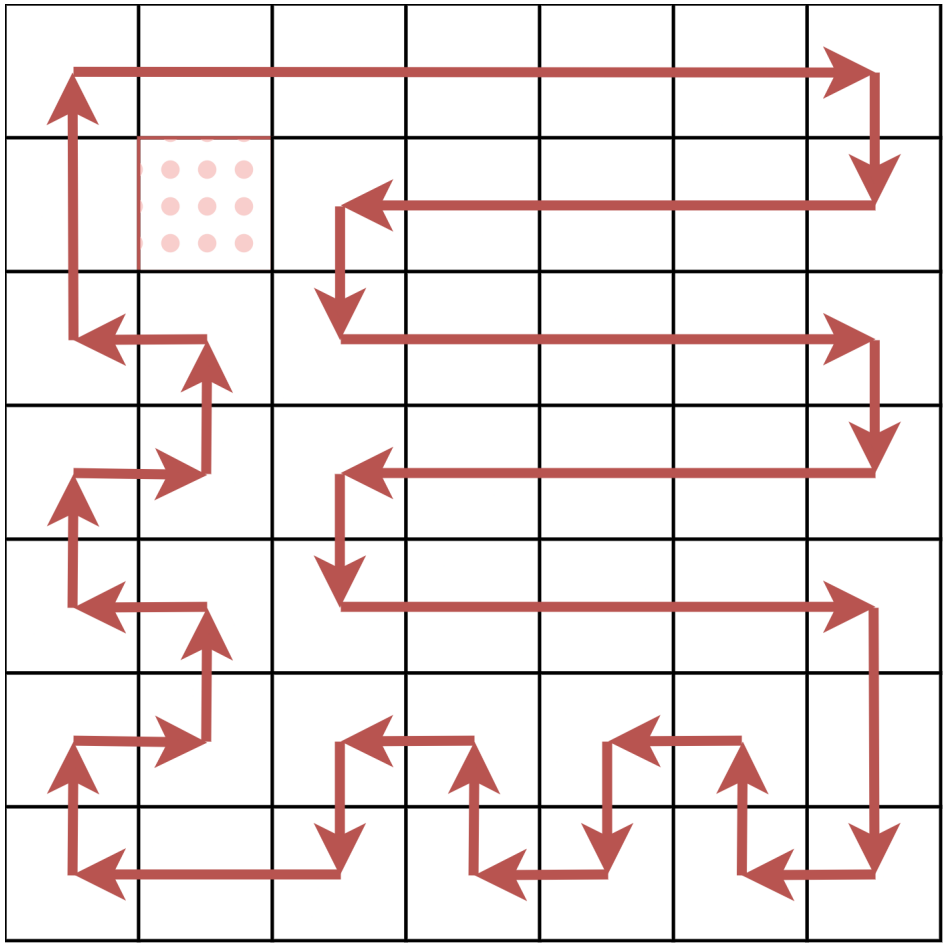
G. Snake

- 如果行或列是偶数，可以进行下图的构造（以行是偶数为例）



G. Snake

- 如果行和列均是奇数，可以进行下图的构造



- 可以看到图里没经过 $(2,2)$，但是可以在 $(2,1)$ 选择要去 $(1,1)$ 还是 $(2,2)$，再从 $(1,2)$ 离开，仍然没问题。

H. Telepathy

题意

- 通信题。给定一个 1 到 n 的集合 S，满足 $|S| < \frac{n}{2}$。
- First Run：选一个不在 S 里的数加进去，得到 S'
- Second Run：得到打乱后的 S'，反推出之前加进去的数。
- $n \leq 10^6$

H. Telepathy

- 将 1 到 n 排成一个序列，在集合中的元素视为左括号，不在集合中的视为右括号，从左到右贪心做括号匹配。
- 对原始集合 S，有 k 个左括号和 $(n-k)$ 个右括号。因为 $k < \frac{n}{2}$，所有左括号都能被匹配，剩下 $(n-2k)$ 个未匹配的右括号。
- First Run 的策略：选择最后一个未匹配的右括号 x 加入集合。复杂度 $O(n)$。
- Second Run 的策略：对收到的 S' 做同样的括号匹配，输出栈底。复杂度 $O(n)$。

H. Telepathy

正确性

- First Run 中因为 x 未匹配，说明扫描到 x 时栈为空。Second Run 在 x 之前的序列与 First Run 完全相同，故 Second Run 扫描到 x 时栈也为空，所以 x 被推入栈底。
- x 是最后一个未匹配右括号，因此在原序列中 x 之后的所有左右括号恰好完美配对。Second Run 的序列在 x 之后也与 First Run 相同，这些元素自行匹配，不会弹出栈底的 x。

I. Calendar Cubes

题意

- 立方体的六个面上分别写一个 $0-8$ 的整数（6 可用作 9），这样两个立方体就可以展示一个两位数。
- 定义两个立方体不能表示的最小正整数为它们的 MEX。
- 询问是否存在一组构造使得 MEX $= x$ 。多组询问。
- $1 \le T \le 99,\ x \in [01, 99]$ 。

I. Calendar Cubes

- 单个立方体的赋值方案可以看作是包含六个数字的可重集。
- 插板法可得不同的方案数为

$$\binom{9+6-1}{6} = 3003$$

- 因此两个立方体方案数不超过 10^7，暴力枚举预处理即可。
- 时限足够程序枚举 + 输出，本地直接打表提交也完全可以。。事实上有解的情况并不多。

J. Crossroads

题意

- n 条竖线 $x = a_i$，m 条横线 $y = b_j$。沿竖线 i 移动单位距离花费 t_i，沿任意横线花费 t_0。
- $f(i, j)$ 为从 $(0, 0)$ 到 (a_i, b_j) 沿网格线移动的最短时间。
- q 次查询 (l, r, d, u)，求 $\sum_{i=l}^{r} \sum_{j=d}^{u} f(i, j) \bmod 998244353$。
- $n, m, q \leq 2 \times 10^5$。

J. Crossroads

贪心观察

- 所有横线代价相同，因此最优路径只需经过一条竖线 k：沿 $y=0$ 走到 a_k，沿 $x=a_k$ 走到 b_j，沿 $y=b_j$ 走到 a_i。
- 若 $k \le i$：代价 $= t_k b_j + t_0 a_i$，与 a_k 无关，取 $m_i = \min_{k \le i} t_k$。
- 若 $k > i$：代价 $= t_k b_j + 2t_0 a_k - t_0 a_i$，需要同时优化 t_k 和 a_k。

公式整理

- 提公因式 $-t_0 a_i$，得

$$f(i,j) = \min\left(m_i b_j + 2t_0 a_i,\ \min_{k>i}(t_k b_j + 2t_0 a_k)\right) - t_0 a_i$$

- 每个候选 k 对应一条关于 b_j 的一次函数 $t_k x + 2t_0 a_k$。$\min_{k>i}$ 即下凸壳。
- 左侧选项截距 $2t_0 a_i$ 最小，在小 b_j 处占优；存在分界点 pos 使得前缀由左侧决定，后缀由凸壳决定。

J. Crossroads

扫描线 + 凸壳

- 从 $i = n$ 到 1 扫描。每步将直线 $(t_i, 2t_0a_i)$ 加入凸壳。
- 由于 a_i 递减，新直线截距最小；凸壳中斜率 $\geq t_i$ 的旧线被完全淘汰，新线成为凸壳最左段。更新只需弹栈 + 入栈。
- 查询 $\sum_{i=l}^{r}$ 拆为后缀和之差：在 $i = l$ 处查询（加），$i = r+1$ 处查询（减）。

线段树

- 以 j 为下标建线段树，维护 $\mathrm{Ans}[j]$（答案累加器）和 $C[j]$（凸壳当前值）。
- 每步 i：对 $[1, \mathrm{pos}]$ 累加 $m_i b_j + 2t_0 a_i$，对 $[\mathrm{pos}+1, m]$ 累加 $C[j]$，全局减 $t_0 a_i$。
- 三类懒标记：Ans 加法 $(kb_j + b)$、Ans 加 C（次数 cnt）、C 覆盖赋值 $(kb_j + b)$。
- 下传顺序：先 Ans 加法，再 Ans 加 C，最后 C 覆盖。确保加 C 时使用的是覆盖前的值。

J. Crossroads

复杂度

- 凸壳总插入 n 条线，每条至多弹出一次，总计 $O(n)$。
- 每步 i 做 $O(\log m)$ 的线段树操作和凸壳上二分。
- 每个查询 $O(\log m)$。
- 总时间复杂度 $O(n \log m + q \log m)$。

K. Sum and Product

题意

- 令 $f(n,k)$ 表示满足 $k(a+b) \equiv ab \pmod{n}$ 的整数对 (a,b) 的数量 $(0 \le a,b < n)$。
- 给定 n 和 m，求 $\sum_{k=0}^{m} f(n,k)$
- $n \le 10^{14}, m < n$

$$
\begin{aligned}
ka + kb &\equiv ab \pmod{n} \\
ab - ka - kb &\equiv 0 \pmod{n} \\
ab - ka - kb + k^2 &\equiv k^2 \pmod{n} \\
(a-k)(b-k) &\equiv k^2 \pmod{n}
\end{aligned}
$$

- 因为 a, b 在 $[0,n)$ 内任取，所以 $(a-k) \bmod n$ 和 $(b-k) \bmod n$ 也在 $[0,n)$ 内任取。
- 所以问题变为：求有多少整数对 (x,y) 满足 $0 \le x,y < n$ 且 $xy \equiv k^2 \pmod{n}$。

K. Sum and Product

- 对于固定的 x，一次同余方程 $xy \equiv M \pmod{n}$ 有解的充要条件是 $\gcd(x, n) \mid M$，且在 $[0, n)$ 内恰有 $\gcd(x, n)$ 个解。
- 因此考虑枚举 n 的因数 d，设有 u_d 个 x 满足 $\gcd(x, n) = d$，有 v_d 个 k 满足 $d \mid (k^2 \bmod n)$，那么答案就是 $\sum_d d u_d v_d$
- 令 $x = cd$，为了让 $\gcd(cd, n) = d$，必须有 $\gcd(c, \frac{n}{d}) = 1$。这就是欧拉函数 $\phi(\frac{n}{d})$ 的定义。所以 $u_d = \phi(\frac{n}{d})$。
- 根据 gcd 的性质（辗转相除法），$d \mid (k^2 \bmod n)$ 等价于 $d \mid k^2$。如果 d 有因数 p^t，那么 k 必须要被 $p^{\lceil \frac{t}{2} \rceil}$ 整除。令 $r(d) = \prod_p p^{\lceil \frac{t}{2} \rceil}$ 所以 $v_d = \lfloor \frac{m}{r(d)} \rfloor + 1$。
- 通过 DFS 枚举 n 的所有因数，枚举的过程中顺便维护 ϕ 和 r 即可。复杂度 $O(\sqrt{n})$。

L. Critical Strike

题意

- 有 n 件装备，第 i 件装备暴击率为 p_i，暴击倍数为 v_i，价格为 w_i。若多件装备同时暴击，只有最高的倍数有效。
- 能买总价最高为 m 的装备，每件装备最多买一次，求最大期望值。
- $n, m \leq 3000$

- 01 背包问题的变种。所有装备按倍率从低到高排序，设考虑前 i 件装备且一共花了 j 金钱后，最大期望值为 $f(i,j)$，则有 dp 方程

$$f(i,j) = \max\left(f(i-1,j), f(i-1,j-w_i) \times (1-p_i) + p_i v_i\right)$$

- 复杂度 $O(nm)$。

M. Night at the Museum

题意

- 给定 n 个点构成的闭合折线表示巡逻路线，保安沿 $1 \to 2 \to \cdots \to n \to 1 \to \cdots$ 的顺序巡逻。当保安不在线段端点时，其视野是半径为 r、圆心角为 $2a$ 的扇形，且视野中线方向与当前巡逻线段的前进方向相同。给定 m 件展品的位置，对于每个 $0 \le k \le m$，求能够看到恰好 k 件展品的巡逻路线的总长度。
- $3 \le n \le 2 \times 10^3$，$1 \le m \le 2 \times 10^3$。

M. Night at the Museum

- 分别考虑每条巡逻线段，展品 Q 在线段上某点 P 的视野内，相当于点 P 在展品 Q 的“反向视野”内。
- 展品 Q 的“反向视野”边界所在的直线和圆与巡逻线段求交点，将巡逻线段分成若干段，每段取中点判断是否在展品 Q 的“反向视野”内，可以求出巡逻线段上能看到展品 Q 的线段区间。
- 枚举展品可以得到 $O(m)$ 个线段区间，扫描线一次就能对每个 k 统计出恰好被恰好 k 个线段区间覆盖的线段长度总和。
- 时间复杂度 $O(nm \log m)$。

Thank you!