2024 ICPC 国际大学生程序设计竞赛
亚洲区域赛（沈阳站）

华为云计算与网络创新 Lab

2024 年 11 月 24 日

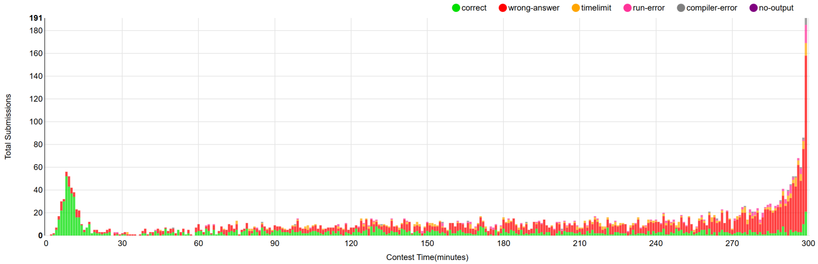
概况

- Easy： J
- Medium-Easy： B、D、E、M(+)
- Medium： A(+)、G(−)、H、I
- Medium-Hard： C、F、K、L

概况 (cont'd)

Attempted
Accepted
320
300
250
200
150
100
50
0
# of Teams
A
B
C
D
E
F
G
H
I
J
K
L
M
Problem

概况 (cont'd)



correct
wrong-answer
timelimit
run-error
compiler-error
no-output
Total Submissions
191
180
160
140
120
100
80
60
40
20
0
0
30
60
90
120
150
180
210
240
270
300
Contest Time(minutes)

J. Make Them Believe

题意

- 给定参加八强单败淘汰赛的八支战队的名称和实力值，找出冠军和亚军队伍的名称。

- 赛中通过：320 支队伍，封榜前 320 支队伍（所有队伍）。
- 首次通过：北京理工大学珠海学院 今天是人类真正幸福的光辉历史的转折点（打星队伍） 3 分钟。
- 正式队伍首次通过：华中科技大学 叽里咕噜的说什么呢 4 分钟。

- 决赛的两支战队分别是前四支战队和后四支战队里实力值最高的，再比较这两支战队的实力值即可，也可以直接按照题意模拟赛程。

E. Light Up the Grid

题意

- 有一个 2×2 的网格，每个格子可以处于开启或关闭状态，但是不知道每个格子的实际状态。给出 4 种切换格子状态的操作，每种操作可以切换特定形状的子网格的格子状态，对应代价分别是 a_0, a_1, a_2, a_3。要构造一个总代价最小的操作序列，使得无论起始网格是给定的 m 种网格中的哪一种，总能在某次操作后所有格子都是开启状态。T 次询问，每次给出起始网格的集合。
- $1 \leq m \leq 16$，$1 \leq T \leq 10^4$。

- 赛中通过：188 支队伍，封榜前 169 支队伍。
- 首次通过：北京大学 赤橙黄绿蓝紫 23 分钟。

E. Light Up the Grid (cont'd)

- 记 S 表示未达到过全开启状态的可能网格的集合，每种操作会将 S 中的网格 u 变成 u'，得到 u' 的集合为 S'，再从 S' 中去除全开启状态的网格。
- 对所有网格的集合按照上述转移建一张图，转化成每次游戏给定一个集合 S，求 S 到 $\varnothing$ 的最短路，只需要在反图上以 $\varnothing$ 为源点跑一次最短路就可以求出所有 S 的答案。

D. Dot Product Game

题意

- 给两个 1 到 n 的排列 A 和 B，Alice 和 Bob 轮流操作，Alice 先操作，Alice 只能操作排列 A，Bob 只能操作排列 B，每次操作可以选两个元素交换，使得点积严格变大，不能操作的输，问最优策略下谁赢。
- 另有 $(n-1)$ 次修改操作，每次选取其中一个排列的区间 $[l, r]$ 做 k 次区间左移。
- $1 \le n \le 5 \times 10^5$。

- 赛中通过：180 支队伍，封榜前 145 支队伍。
- 首次通过：北京大学 赤橙黄绿蓝紫 39 分钟。

D. Dot Product Game (cont'd)

- 对于两个下标 i 和 j，能交换 a_i 和 a_j 或者交换 b_i 和 b_j 当且仅当 $a_i b_i + a_j b_j < a_i b_j + a_j b_i$，也就是 $(a_i - a_j)(b_i - b_j) < 0$，可以改写成 $(i' - j')((ba^{-1})_{i'} - (ba^{-1})_{j'}) < 0$，这意味着排列 BA^{-1} 的逆序数会减少。

D. Dot Product Game (cont'd)

- 观察到交换 A 或者 B 中的两个不同元素，对 BA^{-1} 的影响也是交换两个不同元素，同时会改变逆序数的奇偶性。游戏结束当且仅当逆序数为 0，因此操作次数的奇偶性一定是逆序数的奇偶性，也是 n 减去排列 BA^{-1} 的置换环个数的奇偶性，可以在 $O(n)$ 的时间复杂度下求解。
- 对于修改操作，区间 $[l, r]$ 左移可以用 $(r-l)$ 次交换操作来实现，只需关注 $k(r-l)$ 的奇偶性，每次修改可以直接 $O(1)$ 回答，并不需要实际对排列进行操作，总时间复杂度是 $O(n)$。

B. Magical Palette

题意

- 给定 n 和 m，构造 $a_1, a_2, \ldots, a_n$ 和 $b_1, b_2, \ldots, b_m$ $(0 \le a_i, b_j < nm)$ 使得 $a_i b_j \bmod nm$ 互不相同。
- $1 \le n, m, n \times m \le 10^6$。

- 赛中通过：165 支队伍，封榜前 120 支队伍。
- 首次通过：浙江工业大学 下次还填非常简单 38 分钟。

B. Magical Palette (cont'd)

- 不妨设 $a_1 b_1 \bmod nm = 0$，那么存在 $uv = nm$ 使得 $u|a_1$ 且 $v|b_1$，此时第 1 行最多有 nm/u 个不同的数，但是不能少于 m，因此 $u \le n$，同理 $v \le m$，那么只能是 $u = n$ 且 $v = m$。
- 此时第 1 行有 $0, n, 2n, \ldots, (m-1)n$ 各一个，第 1 列有 $0, m, 2m, \ldots, (n-1)m$ 各一个，那么第 1 行和第 1 列之间有 $\gcd(n, m)$ 个相同的数，只能是 $\gcd(n, m) = 1$，否则无解。

- 对于 $\gcd(n, m) = 1$，构造 $a_i = (im + 1) \bmod nm$ $(1 \le i \le n)$ 和 $b_j = (jn + 1) \bmod nm$ $(1 \le j \le m)$。
- 如果存在 $a_i b_j \equiv a_{i'} b_{j'} \pmod{nm}$，展开整理可以得到 $m(i - i') \equiv n(j - j') \pmod{nm}$，可以知道 $i \equiv i' \pmod{n}$，由于 $1 \le i, i' \le n$，也就是 $i = i'$，同理 $j = j'$。这意味着 $a_i b_j \bmod nm$ 互不相同，也就得到了一组可行解。

M. Obliviate, Then Reincarnate

题意

- 给定 n 和 m，有 n 种操作 (a, b)，每种操作可以从集合 S 中选出 $w \equiv a \pmod{m}$，将 $(w + b)$ 加入到 S 中，q 次询问初始 $S = \{x\}$ 时能否将 S 扩充为无限集。
- $1 \le n, m, q \le 5 \times 10^5$。

- 赛中通过：94 支队伍，封榜前 59 支队伍。
- 首次通过：北京理工大学 越野 49 分钟。

M. Obliviate, Then Reincarnate (cont'd)

- 对于一个操作 (a, b)，连一条 $a \bmod n$ 到 $(a + b) \bmod n$ 的权值为 b 的有向边，可以得到一张 n 个点 m 条边的有向图。对于一个询问 x，如果从 $x \bmod n$ 出发沿着有向边能到达某个权值和非 0 的环，就可以沿着这个环不断操作得到无穷多的数，此时询问的结果为 "Yes"，否则结果为 "No"。

M. Obliviate, Then Reincarnate (cont'd)

- 考虑一个强连通分量，判断其中是否存在权值和非 0 的环。如果所有环的权值和均为 0，可以证明存在给每个点 u 赋值一个势能 h_u 的方案，使得每条边 $u \to v$ 的权值恰好是 $(h_v - h_u)$。通过 DFS 或者 BFS 给所有点赋值，检查每条边是否满足条件即可。
- 对原图做强连通分量缩点得到一个 DAG，通过拓扑排序或者记忆化 DFS 可以求出从每个分量出发能否到达某个有权值和非 0 的环的分量（包括自己），每次询问只需要检查 x 所在分量，总时间复杂度是 $O(n + m + q)$。

G. Guess the Polygon

题意

- 乱序给出一个简单多边形的 n 个顶点，每次可以询问竖直线 $x = p/q$ 与简单多边形的相交部分总长，交互器以分数形式 r/s 返回结果，要在 $(n-2)$ 次询问内求出简单多边形的面积，以分数形式 u/v 输出。
- $3 \le n \le 1\,000$，坐标是 $[0, 1\,000]$ 内的整数。

- 赛中通过：47 支队伍，封榜前 27 支队伍。
- 首次通过：浙江大学 来生 43 分钟。

- 记 $f(x)$ 表示 x 处的竖直线与简单多边形的相交部分总长，那么就是计算 $\int_0^{1\,000} f(x)dx$，或者简单理解为 x 轴上方折线 $f(x)$ 下方的面积。
- 观察到 $f(x)$ 是分段线性函数，分段点是简单多边形各顶点的横坐标，只考虑 $f(x) > 0$ 的段，接下来分情况讨论：
  - 各顶点的横坐标互不相同，此时 $f(x)$ 是 $(n-1)$ 段的连续分段函数，询问 $(n-2)$ 个分段点处的函数值即可；
  - 否则 $f(x)$ 是至多 $(n-2)$ 段的分段函数，询问每一段中点处的函数值即可。

G. Guess the Polygon (cont'd)

- 需要特别注意的是交互器返回的 r 和 s 可能都很大，这里有一些处理方法：
  - 观察到简单多边形面积的两倍是不超过 2×10^6 的整数，且 s 不存在超过 $1\,000$ 的质因子，因此可以任意选取一个大于 2×10^6 的质数 p，分数对 p 取模进行计算；
  - 选取合适的编程语言和数据类型：C++ 可以用 long double 类型读入和计算，结果四舍五入到整数，精度是足够的；Python 可以用分数类型或者浮点类型做计算。

题意

- 给出一棵有 n 层边的满二叉树的所有边的边权，初始这棵树只有根，接下来 n 天，每天上午会新生长出一层，每天下午可以选择至多一条边将其边权修改为任意正整数，要修改最少的边使得 n 天后根到每个叶子的简单路径的边权和两两不同，或判定无解。
- $1 \leq n \leq 10$。

- 赛中通过：19 支队伍，封榜前 10 支队伍。
- 首次通过：北京大学 呆呆鸟 56 分钟。

I. Growing Tree (cont'd)

- 观察到限制就是对每个 $k = 1, 2, \ldots, n$，最深的 k 层边可以修改不超过 k 条，因此修改的边深度越小，限制会越宽松。
- 考虑最深的点 u 使得 u 子树中有两个叶子深度相同，那么 u 向下的两条边一定要修改一条，并且如果两条边都修改，可以将其中一条边换成修改 u 向上的边，仍然能满足限制，因此只需要修改其中一条。
- 枚举修改两条边之中的一条然后递归处理，由于最多修改 n 条边，递归次数是 $O(2^n)$，每次找 u 的时间复杂度是 $O(n2^n)$，总时间复杂度是 $O(n4^n)$。

H. Guide Map

题意

- 给定一张 n 个点的完全图，给定 $(n-2)$ 条特殊边，这些特殊边构成两棵树，计算完全图中选取边的子集的方案数，使得从 1 出发按照规则遍历边导出子图时可以经过所有特殊边，同时只经过不多于一条不同的非特殊边。答案对 $998\,244\,353$ 取模。
- $2 \le n \le 2 \times 10^5$。

- 赛中通过：16 支队伍，封榜前 8 支队伍。
- 首次通过：北京大学 赤橙黄绿蓝紫 54 分钟。

H. Guide Map (cont'd)

- 遍历规则实际上就是 DFS，在多个未访问的相邻点中选择标号最小的访问，得到一棵要包含所有特殊边的 DFS 树。
- 考虑经过的非特殊边 (p, q)，这条边连接了两棵树之后得到了一棵树。
- 对于一棵树的情况，以 1 为根，可以知道对于一条树边 $u \to v$，u 除了一定要和 v 连边之外，还可以和 v 子树内标号 $> v$ 的点任意连边，其他点对之间连边都不符合要求。记可以任意连边的点对数为 k，方案数就是 2^k。
- 注意当 1 所在的树大小为 $(n-1)$ 时可以不经过特殊边，此时也按照一棵树的情况处理。

- 不难发现添加不同的非特殊边 (p, q) 不会得到相同的方案，只需对所有 (p, q) 求和。
- 添加 (p, q) 之后：
  - T_1 内的可任意连边点对不受影响；
  - T_2 内的可任意连边点对的计算相当于一棵树的情况里 T_2 的根是 q，这部分和 p 无关；
  - T_1 和 T_2 之间的可任意连边点对在 T_1 里 1 到 p 的路径上的点和 T_2 里的点之间产生，这部分和 q 无关。
- 后两部分可以按照 DFS 序分别遍历 p 和 q 进行计算，使用乘法原理得到答案。
- 其中涉及到一些子树内外的值域数点，按照 DFS 序将树展开成序列之后子树内外都是 $O(1)$ 段区间，转化成二维数点问题，使用可持久化线段树维护的时间复杂度是 $O(n \log n)$。

A. Safety First

题意

- 一个 n 段的梯子里每段的长度是递减的，如果对每个 $i = 1, 2, \ldots, n-1$，第 $(i-1)$ 段的上端和第 i 段的上端或者下端锁定，这个梯子就是“稳定的”，计算 n 段的高度为 m 的“稳定的”梯子数量。T 次询问，答案对 $998\,244\,353$ 取模。
- $1 \leq n, m \leq 2\,000$，$1 \leq T \leq 10^5$。

- 赛中通过：2 支队伍，封榜前 2 支队伍。
- 首次通过：北京大学 赤橙黄绿蓝紫 160 分钟。

A. Safety First (cont'd)

- 观察到第 1 段也就是最长段的长度一定对梯子高度有贡献，而对每个 $i = 2, 3, \ldots, n$，如果第 $(i-1)$ 段的上端和第 i 段的下端锁定，那么第 i 的长度也会对梯子高度有贡献，这意味着除了最长段之外的每一段都可以选择是否要将其长度贡献给梯子高度。

A. Safety First (cont'd)

- 一个做法是 $dp1_{i,j,k}$ 表示梯子已经有 i 段，从大到小考虑到长度为 j 的段，梯子高度是 k 的方案数，转移是要么新增一个长度为 j 的段，如果不是第一段还要决策是否将其长度贡献给梯子高度，要么是转而考虑长度为 $(j-1)$ 的段。这个做法的时间复杂度是 $O(nm^2)$，难以通过。
- 另一个做法是 $dp2_{i,j,k}$ 表示梯子已经有 i 段，其中 j 段对梯子高度有贡献，梯子高度是 k 的方案数，转移是要么新增一个长度为 1 的段，如果不是第一段还要决策是否将其长度贡献给梯子高度，要么已有的段长度都加 1。这个做法的时间复杂度是 $O(nm\min(n,m))$，也难以通过。

A. Safety First (cont'd)

- 考虑到在 $dp2$ 中，如果每一段的长度都至少是 $\sqrt{m}$，那么对梯子高度有贡献的段数不会超过 $\sqrt{m}$，因此可以在 $dp2$ 中要求每次新增的段长度是 $\lceil\sqrt{m}\rceil$ 而不是 1，然后使用 $dp1$ 处理长度小于 $\lceil\sqrt{m}\rceil$ 的段，从而得到时间复杂度是 $O(nm\sqrt{m})$ 的做法。预处理所有答案之后每次可以 $O(1)$ 回答询问。

F. Light Up the Hypercube

题意

- 有一个 2^n 个顶点的 n 维超立方体，每个顶点可以处于开启或关闭状态，但是不知道每个顶点的实际状态。给出 2^n 种切换顶点状态的操作，每种操作可以切换特定形状的子立方体的顶点状态，对应代价分别是 $a_0, a_1, \dots, a_{2^n-1}$。要构造一个总代价最小的操作序列，使得无论起始各顶点状态如何（有 2^{2^n} 种可能），总能在某次操作后所有顶点都是开启状态。答案对 $998\,244\,353$ 取模。
- $1 \le n \le 20$。

- 赛中通过：2 支队伍，封榜前 2 支队伍。
- 首次通过：北京大学 呆呆鸟 196 分钟。

- 将子立方体的顶点状态的抽象成 2^{2^n} 个点，操作抽象成操作前状态连向操作后状态、权值为代价的边。
- 答案等于遍历所有点至少一次的最短路。

F. Light Up the Hypercube (cont'd)

- 鉴于该图的高度对称性，考虑答案的下界。
- 一个保守的猜测是出发点不计首次经过的哈密顿路，即走了 2^{2^n} 条边、遍历所有点恰好一次。
- 一个大胆的猜测是最小生成树额外加上一条权值最小的边。
- 下面归纳证明后者可以构造出相应的路。

F. Light Up the Hypercube (cont'd)

- 根据 Kruskal 算法，我们贪心地优先插入将代价小的操作。
- 当前森林的每棵树中，任取两点的状态位集异或所得值构成线性空间。
- 我们假想存在一个线性基维护这个线性空间，每次插入操作时将操作可选的各子立方体的反转位集加入线性基。
- 通过线性基的大小变化可以求出当前插入操作要用多少次。

F. Light Up the Hypercube (cont'd)

- 按照这个思路，回到 n 维超立方体，取任一点作原点。
- 对于所有当前插入操作可选的子立方体，如果离原点顶点最远的顶点未被标记，标记之（因为其状态在线性基中可以留给低位表示）。
- 由于存在单点操作，最终所有顶点均被标记。
- 逆向考虑这个过程，哈密顿路第一步走权值最小的边，利用格雷码的思想每次将哈密顿路复制一份、纳入标记的顶点构造出答案。

F. Light Up the Hypercube (cont'd)

- 现在设计算法，实则核心代码不超过五行。
- 贪心地按代价排序，操作 i 能标记到所有超集（$j \wedge i = i$）顶点 j（如果尚未标记）。
- 应用高维前缀 $\min$，根据每个顶点被哪个操作标记统计答案，时间复杂度 $O(n2^n)$。

L. The Grand Contest

题意

- 给出一场比赛两支队伍的 n 次提交，按照规则对这两支队伍排名，问是否存在区间 $[L, R]$ 使得从比赛中“移除”这个时间区间之后，两支队伍排名交换，找出 $(R - L)$ 最小（相同取 L 最小）的区间，或判定无解。
- $1 \leq n \leq 4 \times 10^5$。

- 赛中通过：1 支队伍，封榜前通过 1 支队伍。
- 首次通过：北京大学 赤橙黄绿蓝紫 191 分钟。

L. The Grand Contest (cont'd)

- 模拟出两支队伍的过题数和解题总时间，由于“移除区间”不能改变过题数，如果题数不同就无解，否则可以知道落后队伍要追回领先队伍多少时间差，记为 K。
- 记 $cnt[L, R]$ 表示 $[L, R]$ 时间内的过题数之差（落后队伍的减去领先队伍的），$sum[L, R]$ 是过题时间之差，考虑每个时间段的影响：
  - $[L, R)$ 时间内的过题时间 X 都会变成 L，也就是贡献 $(X - L)$，这部分追回的时间差是 $(sum[L, R) - L \times cnt[L, R))$；
  - $[R, +\infty)$ 时间内的过题时间都会减去 $(R - L)$，这部分追回的时间差是 $(R - L) \times cnt[R, +\infty)$；
- 那么总共追回的时间差可以整理成 $(f(L) - f(R))$，其中 $f(T) = sum[T, +\infty) - T \times cnt[T, +\infty)$。

L. The Grand Contest (cont'd)

- 现在要求出使得 $f(L) - f(R) \geq K$ 的 $(R - L)$ 最小（相同取 L 最小）的解，可以证明在允许实数解的时候最优解必然有 L 或者 R 恰好是某次过题的时间，在要求整数解的时候，对非整数一侧做相应取整仍然可以得到 $(R - L)$ 最小的解，再尝试向左平移得到 L 最小的解。
- 因此做法是枚举一个过题时间作为 L，在 RMQ 或者单调栈上二分出最小的过题时间 R' 使得 $f(L) - f(R') \geq K$，在 R' 和前一次过题时间之间解不等式得到最小的 R，再尝试向左平移这组 $[L, R]$，类似地枚举一个过题时间作为 R 也做一次，时间复杂度是 $O(n \log n)$。

C. Crisis Event: Meteorite

题意

- 有 n 个格子排成一排，初始某些格子里有角色。接下来有 m 轮，第 i 轮开始时第 j 个格子会新落下 $a_{i,j}$ 块陨石（可能为 0），如果有角色被陨石砸到那么游戏直接失败。在第 i 轮结束前，角色可以任意多次移动到相邻的格子里，角色之间可以重合，但是如果要走到的格子里有陨石就要把格子里已有的陨石全部挖走。计算通过这 m 轮所需挖走的最少陨石总数，或判定无解。
- $1 \le n, m, n \times m \le 10^6$。

- 赛中通过：1 支队伍，封榜前 0 支队伍。
- 首次通过：北京大学 呆呆鸟 272 分钟。

- 无解的情况只有第 1 轮开始时就有角色被陨石砸到，或者某一轮中所有格子均有陨石落下，接下来只考虑有解的情况。
- 每个格子都只关心最后一次进入的轮次 k，这个格子在第 k 轮及之前的陨石都要挖走，第 k 轮之后的陨石都不用挖走。
- 对于只有一个角色的情况，考虑时光倒流，从角色在第 m 轮的位置开始回溯时间，走过的格子是一个区间，这些区间里的格子里更早轮次的所有陨石都已经被挖走。
- $dp_{i,l,r}$ 表示到时光倒流第 i 轮时这个角色走过了 $[l, r]$ 区间里的格子所需挖走的最少陨石总数，m 轮下来的总状态数是 $O(nm^2)$，不能接受。

C. Crisis Event: Meteorite (cont'd)

- 对于一个区间 $[l, r]$，如果更早一轮在 $[l, r]$ 里有格子不会落下陨石，就可以直接走到这些格子，否则只需要考虑走到区间两侧最近的不会落下陨石的格子，最多有两个转移。
- 初始有效的区间都是单点，可以证明所有有效的区间 $[l, r]$ 满足 l 非降的同时 r 也非降，只有 $O(n)$ 个有效区间，m 轮下来的总状态数是 $O(nm)$，可以接受。
- 在每一轮中，所有转移按照 $[l, r]$ 双关键字基数排序，再用双指针对结果进行聚合，就可以做到 $O(n)$ 的时间复杂度。

C. Crisis Event: Meteorite (cont'd)

- 现在初始有多个角色，不难发现如果有一些角色要相遇，只需要在第 1 轮相遇，这样会分成若干个区间，每个区间里的角色走到某一个格子，在之后的轮次里作为一个角色行动。
- 选取前一部分得到的 $O(n)$ 个区间之中的若干个，每个区间内的角色可以不用再走，区间外的角色需要向左或者向右走到一个区间，并沿途挖空第 1 轮落下的陨石。
- $dp_{i,0/1/2}$ 表示考虑到第 i 个格子，当前格子没有区间外的角色经过 / 有区间外的角色向左经过 / 有区间外的角色向右经过时所需挖走的最少陨石总数。如果遇到候选的区间，可以直接在左端点走上区间，使用单调队列维护区间每个点走下区间的转移，这部分的时间复杂度是 $O(n)$。
- 总时间复杂度是 $O(nm)$。

K. Fragile Pinball

题意

- 给定一个有 n 条边的凸多边形，有一个沿直线运动的弹球，如果遇到边界时可以选择做反射，如果同时在两条边上可以选择做反射边界及其顺序，但是每条边只能反射一次弹球。对每个 $k = 0, 1, \ldots, n$ 计算弹球被反弹至多 k 次时，弹球在凸多边形内可以行进的最长路程。
- $3 \leq n \leq 6$。

- 赛中通过：0 支队伍。

K. Fragile Pinball (cont'd)

- 枚举和弹球发生碰撞的边界及其顺序，按照碰撞的顺序将起始多边形做一系列对称得到一个目标多边形，这相当于将弹球的路径展开成一条直线，这样弹球需要在起始多边形里找一个起点，目标多边形里找一个终点，使得起点和终点之间的连线穿过所有沿途反射边。

K. Fragile Pinball (cont'd)

- 可以通过平移和旋转的微扰证明一定存在最优解满足弹球经过以下至少两个不同的点：
  - 起始多边形的顶点；
  - 目标多边形的顶点；
  - 沿途反射边的端点。
- 枚举两个不同的点，连成的直线作为弹球运动的轨迹，直线与起始多边形和目标多边形求交可以分别得到起点和终点，这样相当于得到了弹球的路径，只需再验证路径是否合法。

Thank you!