2026 年国际大学生程序设计竞赛 全国邀请赛（陕西）

Cfz 出题组

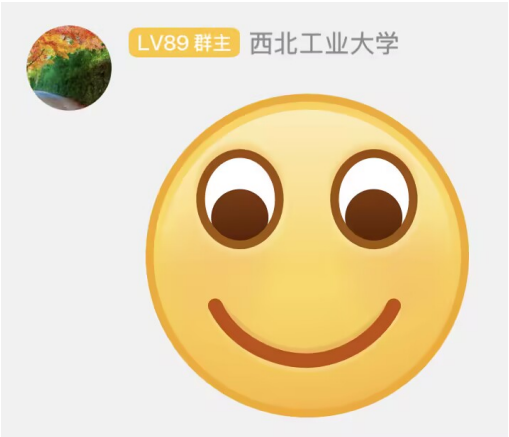
2026 年 5 月 2 日

预期难度 & 预期铜银金线

- E<J<AD<B<CG<KI<N<FL<H<M
- 铜 4 银 6 金 7

一些彩蛋

- 去年区域赛的题目首字母是 A 到 M，今年邀请赛的题目首字母是 N 到 Z。
- 去年区域赛 L 题和今年邀请赛 J 题说的都是同一个人。是谁呢?
- 热身赛题面首页学习了一下深圳邀请赛，放了下面的图，虽然我做这个 Beamer 的时候并不知道主办方有没有原样打印下来（



一些说明

- 听说隔壁赛区要引入子任务了，所以实验了一下，在这场比赛放了一道 easy version & hard version 题目，希望大家能多多包容并提提意见！
- 因为时间比较紧张，所以题解写的比较简略，有问题可以私信我 qwq（QQ 3759713328）
- 请大家不要公开转发题目与题解！
- WBG 深渊夺冠谢谢。

E. Registration 题意

- 有 n 支队伍，第 i 支队伍会在第 v_i 秒访问报名网页。
- 对于每个正整数 s，若在第 s 秒时访问报名网页的队伍不大于 x，则这些队伍报名成功，否则这些队伍报名失败。
- 求报名成功的队伍数量。
- $1 \le n \le 2 \cdot 10^5$，$1 \le x \le n$，$1 \le v_i \le 10^9$，$\sum n \le 2 \cdot 10^5$。

E. Registration 做法

- 开一个 map 记录每一秒时访问报名网页的队伍数量，再遍历数组判断每个队伍是否报名成功即可。

J. Would You Make a Convex? 题意

- 有 n 根木棍，第 i 根木棍的长度为 a_i。
- 选出尽可能多根木棍，使得从这些木棍中选出任意大于等于 3 根木棍都能形成凸多边形。
- 构造方案或报告无解。
- $3 \le n \le 5 \cdot 10^5$，$1 \le a_i \le 10^9$，$\sum n \le 5 \cdot 10^5$。

J. Would You Make a Convex? 做法

- 显然，只要选出的最长的木棍的长度小于选出的最短的木棍与次短的木棍的长度之和，就满足条件。
- 因此，将所有木棍按照 a_i 从小到大排序后，最优的方案中选出的木棍一定形成了一段区间。
- 枚举区间左端点，二分得到最靠右的满足条件的右端点即可。

D. Qenerals 题意

- 第 0 秒时，Yuki 拥有 0 个士兵，占领了 1 个堡垒。地图上还有 n 个未被占领的堡垒，第 i 个堡垒的参数为 a_i。
- 对于每个不大于 m 的正整数 i:
  - 第 i 秒开始时，每个 Yuki 所占领的堡垒都会为 Yuki 生产一个士兵。
  - 第 i 秒结束时，Yuki 可以进行任意次操作；每次操作，Yuki 需要选择一个未被占领的堡垒 j，消耗 a_j 个士兵并占领第 j 个堡垒。
- 求第 m 秒结束时 Yuki 最多能拥有多少个士兵。
- $1 \le n \le 5 \cdot 10^5$，$1 \le m, a_i \le 10^9$，$\sum n \le 5 \cdot 10^5$。

D. Qenerals 题解

- 显然，优先占领 a_i 小的堡垒更优，因此将所有堡垒按照 a_i 从小到大排序后，占领的堡垒一定是一段前缀。
- 从小到大枚举 i 表示要占领前 i 个堡垒。从 $i-1$ 处继承信息后，计算能够占领第 i 个堡垒的最小时间 s，在第 s 秒占领第 i 个堡垒，并额外加上 $(m-s)\cdot i$ 的贡献。
- 将每个前缀的贡献取 $\max$ 即可。

A. North and South 题意

- 给定一个长度为 n 的序列 a。
- 定义一次操作为，选择一个长度为偶数的区间 $[l, r]$，将 $[l, r]$ 中满足 $i - l$ 为奇数的 a_i 的值减少 1，$[l, r]$ 中满足 $i - l$ 为偶数的 a_i 的值增加 1。
- 求使序列 a 中的所有数均相等的最小操作次数，或报告无解。
- $1 \le n \le 10^6$，$0 \le a_i \le 10^{12}$，$\sum n \le 10^6$。

A. North and South 题解

- 进行一次操作后 $\sum a_i$ 的值不变，因此序列 a 的最终值是确定的。
- 从前往后操作，每次只选择 $[i, i+1]$ 进行操作，让序列 a 变成最终值。
- 最后把能合并的区间合并起来。设 $[i, i+1]$ 操作了 v_i 次，那么 $[i, i+1]$ 的贡献就是 $\max(v_i - v_{i-2}, 0)$。

B. Operating Robot 题意

- 在平面直角坐标系上有一个机器人，其初始时位于 $(0, 0)$，你需要让它到达 (x, y)。
- 有一个只包含 `012` 的长度为 n 的操作串，`0` 表示向右移动一步，`1` 表示向上移动一步。
- 你需要将操作串中的所有 `2` 替换为 `0` 或 `1`，机器人会按照替换后的操作串循环移动。
- 求字典序最小的替换方式，使得机器人能够到达 (x, y)。
- $1 \le n \le 10^6$，$0 \le x, y \le 10^{18}$，$\sum n \le 10^6$。

B. Operating Robot 题解

- 因为每执行一次操作，横纵坐标中的一个一定会增加 1，所以只有前 $x+y$ 次执行操作是有效的。
- 设 $s=\left\lfloor\frac{x+y}{n}\right\rfloor$，$k=(x+y) \bmod n$，则前 k 次操作会被执行 $s+1$ 次，后 $n-k$ 次操作会被执行 s 次。
- 枚举前 k 次操作中 0 的数量，容易计算出后 $n-k$ 次操作中 0 的数量，每段操作中都把 0 排到尽可能靠前的位置，取字典序最小的解即可。

C. Palindromic and Balanced 题意

- 定义括号串 $s = s_1 \dots s_n$ 为回文平衡括号串，当且仅当 $s_2 \dots s_{n-1}$ 为回文括号串，且 $s_1 \dots s_n$ 为平衡括号串。特殊地，空串和 () 也为回文平衡括号串。
- 给定一个长度为 n 的括号串 s，求其最长回文平衡括号串子序列的长度。
- $n \le 5 \cdot 10^3$，$\sum n \le 10^4$。

C. Palindromic and Balanced 题解

- 考虑如何生成一个回文平衡括号串：
  - 初始时的括号串为 ()。
  - 每次向中点左侧插入一个 ()，向中点右侧插入一个)(；或者向中点左侧插入一个)(，向中点右侧插入一个 ()。
- 于是可以在找到 s 的最左侧的 (和最右侧的) 后，对中间的部分进行区间 dp。
- 设 $f_{l,r}$ 表示区间 $[l, r]$ 能形成的最长的满足生成条件的子序列的长度；转移时按照生成规则，向左侧和右侧分别加入 $(),)($ 或 $)(, ()$ 即可。

G. Transform 题意

- 给定一个大小为 n 的可重集 $S = \{s_1, \ldots, s_n\}$ 和一个整数 k。
- 定义一次操作为，选择 S 的一个子集 S'（S' 可以为空集），将 S' 从 S 中删除，并将 S' 的 mex 添加到 S 中。
- 求使得 $S = \{k\}$ 的最小操作次数对 998244353 取模的结果。
- $1 \le n \le 5 \cdot 10^5$，$0 \le k, s_i \le 10^9$，$\sum n \le 5 \cdot 10^5$。

G. Transform 题解

- 首先特判一些 corner cases：$n = 1$ 且 $s_1 = k$ 时答案为 0，$\operatorname{mex} S = k$ 时答案为 1，$\operatorname{mex} S > k$ 时答案为 2。
- 手玩一下可以发现，由空集生成一个整数 k 需要 2^k 次操作。那么考虑找出 S 中所有能做出贡献的数 $v_1, \ldots, v_m$，答案即为 $2^k - \sum 2^{v_i}$。
- 什么的数能做出贡献呢？首先大于等于 k 的数肯定无法做出贡献，它们参与操作并没有用。其次如果某个数过多也是不行的，因为它们用不完。
- 将 S 中小于 k 的数从大到小排序，动态记录当前需要多少个数用于操作，再判断每个数是否能做出贡献即可。

I. VIP Coupon 题意

- 商店里出售 n 个商品和 m 张优惠券，第 i 个商品的价格为 a_i，第 j 张优惠券的价格为 b_j，参数为 c_j。
- 购买参数为 v 的优惠券后，下一个购买的物品（包括商品和优惠券）的价格会从 x 变为 $\max(x-v,0)$。
- 优惠券的效果会强制在下一次购买时生效，不能自主选择使用时间。每个物品只能购买至多一次，不能重复购买。
- 求买下所有商品的最小花费。
- $n, m \le 5 \cdot 10^5$，$0 \le a_i, b_i, c_i \le 10^9$。

I. VIP Coupon 题解

- $b_i > c_i$ 的优惠券显然是没用的，忽略这些优惠券。
- 把优惠券看成区间 $[b_i, c_i]$，商品看成区间 $[a_i, +\infty]$。
- 由于优惠券的总优惠额度是一定的，所以我们相当于要把这些区间分成 n 份，使得每一份的面积并之和尽可能大。
- 将所有区间按左端点从小到大排序，维护一个堆记录每一份区间集合的右端点，每次把当前区间接在堆中右端点最小的区间集合中即可。

K. XOR and LCA 题意

- 给定一棵包含 2^n 个结点的树。
- 设 $\mathrm{lca}_r(u, v)$ 表示，以结点 r 为根时，结点 u 与结点 v 的最近公共祖先。
- 求 $\bigoplus_{0 \le u < v < 2^n} \mathrm{lca}_{u \oplus v}(u, v)$。
- $1 \le n \le 21$，$\sum 2^n \le 2^{21}$。

K. XOR and LCA 题解

- $\mathrm{lca}_{u\oplus v}(u,v)=w$，当且仅当 $u,v,u\oplus v$ 在 w 的 3 个不同子树中。
- 考虑枚举每个结点 w，计算结点 w 作为 $\mathrm{lca}_{u\oplus v}(u,v)$ 的次数的奇偶性，即 $u,v,u\oplus v$ 在 w 的 3 个不同子树中的点对数量的奇偶性。
- 注意到这个式子实际等价于，u,v 在 w 的 2 个不同子树中的点对数量的奇偶性。因为如果 $u,v,u\oplus v$ 中有 2 个或 3 个结点在同一个子树中，带来的贡献为偶数；而如果它们都位于不同的子树中，带来的贡献为奇数。
- 直接计算 u,v 在 w 的 2 个不同子树中的点对数量即可。

N. Zebra Crossing 题意

- 给定一棵包含 n 个结点的树，每个结点的颜色为黑色或白色。
- 你有一个跳跃能力 k，表示你跳跃一次可以从结点 x 移动到满足 $\mathrm{dist}(x, y) \leq k$ 的结点 y 上。
- 对于每个 $2 \leq i \leq n$，求出从结点 1 到结点 i 的跳跃过程中，踩到黑色结点的次数的最小值。
- $1 \leq n \leq 5 \cdot 10^5$，$1 \leq k \leq n$，$\sum n \leq 5 \cdot 10^5$。

N. Zebra Crossing 题解

- 考虑把跳跃看成，初始时你有 k 点体力，每走一步体力就减少 1，走到白点时将体力恢复成 k，体力变为 0 时答案增加 1 并将体力恢复成 k。
- 以结点 1 为根，先处理出每个结点 u 子树内最近的白色结点（包括结点 u），设该距离为 d_u。
- 从结点 1 开始从上往下遍历整棵树，记录当前体力 x 与答案 ans。移动到结点 u 时，若 $x \geq d_u$，则可以选择走到子树内最近的白色再走回来，即将 x 赋值为 $\max(x, k - d_u)$。若 x 变为 0，则 ans 增加 1 并将 x 赋值为 k。
- 按照此规则遍历整棵树即可求出答案。

F. Split Sticks 题意

- 有 n 根木棍排成一排，第 i 根木棍的长度为 a_i。
- 定义一次操作为：
  - 选择一根木棍，并将其切成长度均为整数的两部分，其中一部分的长度可以为 0。
  - 将切成的左半部分木棍合并到这根木棍左边的第一个木棍；若其左边没有木棍，则左半部分木棍单独作为一根新的木棍。
  - 将切成的右半部分木棍合并到这根木棍右边的第一个木棍；若其右边没有木棍，则右半部分木棍单独作为一根新的木棍。
  - 删除所有长度为 0 的木棍。
- 求使所有木棍的长度均相等所需的最小操作次数。
- $1 \le n \le 10^6$，$1 \le a_i \le 10^6$，$\sum n \le 10^6$。

F. Split Sticks 题解

- 原问题这个形式直接做不太好做，我们考虑换个角度思考这个问题。
- 考虑将这 n 根木棍依次拼接在一起，此时会存在 $n+1$ 个端点，可以发现一次操作等价于把相邻的两个端点合并为这两个端点形成的区间中的任意一个整点位置，不过开头段和结尾段要特殊处理一下，因为你合并后会重新出现新的首尾端点。
- 考虑枚举最终每根木棍的长度，就能知道最终状态了，初始状态能转移到最终状态当且仅当初始状态的任意两个端点之间的区间至多只有最终状态中的一个端点。

F. Split Sticks 题解（续）

- 暴力枚举 $\sum_{i=1}^{n} a_i$ 的每个因数 check 即可，时间复杂度 $O(nd(\sum a))$。
- 考虑再分析一下这个过程，对于一个长度为 l 的段里面必定不会出现两个端点，因此最终的长度一定在 $[\frac{\max a}{2}, 2\max a]$ 之间，那么我们需要让复杂度最满需要找到一个因数个数尽量多的 $\sum a$ 并且需要找一个含有最多因数个数的区间 $[x, 4x]$，此时 $\sum a = 963761198400$ 是最劣的，在区间 $[460161, 1840644]$ 取到因数个数最大值 882 个，由于此题时限开了很大，并且每次 check 都很难跑到严格 $O(n)$ 的复杂度，因此可以通过此题。

H. Unreachable Land 题意

- 给定三个整数 a, b, m。你需要进行 m 轮操作，第 i 轮操作可以令 $a \leftarrow a \bmod (m - i + 1)$ 或者不进行修改。求 m 轮操作后 $a = b$ 的方案数，答案对 998244353 取模。
- 定义两种方案不同，当且仅当存在 $1 \leq i \leq m$，使得一种方案中第 i 轮进行了修改，而另一种方案中第 i 轮没有进行修改。注意，只要选择执行 $a \leftarrow a \bmod (m - i + 1)$ 即视为进行了修改，不论修改后 a 的值是否变化。
- $0 \leq b < m \leq a \leq 2 \cdot 10^5$，$\sum a \leq 2 \cdot 10^5$。

H. Unreachable Land 题解

- 将操作反转，第 k 轮可以将 b 变成 $k \times c + b$，要求 $b < k$，问最后变成 a 的方案数。
- 称一个操作有效当且仅当 b 的值在操作后发生变化，先考虑只进行有效的操作。
- 记 f_i 代表得到 i 的方案数，设下一次有效操作将 i 变成了 $k \times c + i$，其中 $c > 0$ 和 $k > i$，则有转移 $f_i \to f_{k \times c + i}$。
- 能这么转移是因为得到 i 意味着只进行了 $\leq i$ 的有效操作，而转移时有 $k > i$，两者互不影响。
- 这个转移相当于对公差为 k 的等差数列加上 f_i。

H. Unreachable Land 题解（续）

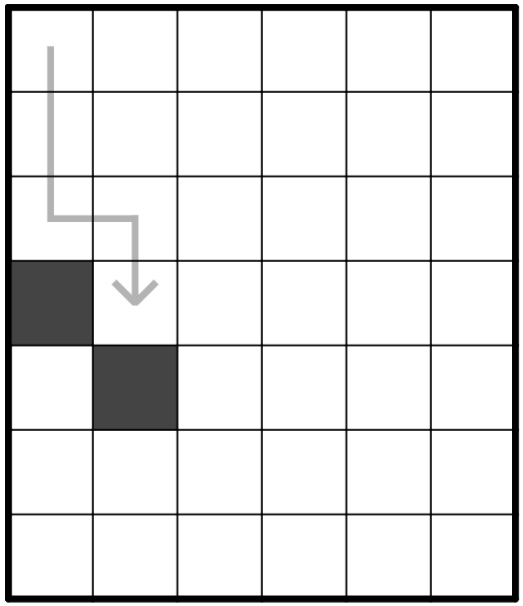
- 对 i 进行根号分治，当 $i \le B$ 时枚举 $k \times c + i$ 暴力转移，这部分时间复杂度为 $O(aB)$。
- 当 $i > B$ 时，$c < \frac{a}{B}$，枚举 c，此时对于这个 c，合法的 k 是一段区间，所以相当于对公差为 c 的等差数列做加法。
- 只需对于每个位置都记录 $\frac{a}{B}$ 个标记，修改时进行差分即可。
- 时间复杂度平衡得到 $O(a\sqrt{a})$。
- 考虑无效操作，对于 $f_i \to f_{k\times c+i}$ 的转移，会有 $k-i-1$ 次操作是无效的，系数为 2^{k-i-1}，带着系数转移即可。
- $i \le B$ 时转移细节较多，需要精细实现避免复杂度多 $\log$。

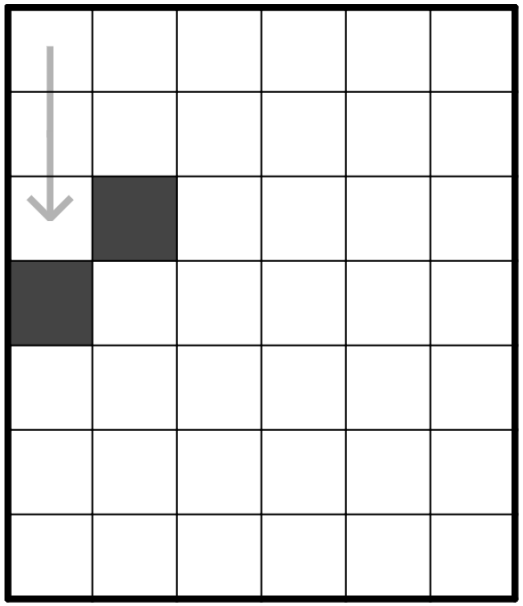
L & M. Yesterday Once More 题意

- 在一个 $n+1$ 行 n 列的棋盘上有 $n-1$ 个障碍，障碍的分布满足：
  - 第 1 行和第 $n+1$ 行中没有障碍。
  - 对于所有 $2 \le i \le n$，第 i 行中有恰好一个障碍。
  - 对于所有 $1 \le j \le n$，第 j 列中有至多一个障碍。
- 初始时你位于 $(1,1)$，你需要通过若干次移动到达棋盘的第 $n+1$ 行。
- 每次移动，你需要选定上下左右中的一个方向，并向该方向移动一个格子。特殊地，若该格子位于棋盘外或该格子中有障碍，则此次移动不会被执行。
- 你需要构造一个移动序列，使得对于任意满足要求的障碍分布方式，你都到达过棋盘的第 $n+1$ 行。
- Easy Version：你需要保证移动序列长度不超过 $30n$。
- Hard Version：你需要保证移动序列长度不超过 $10n$。
- $2 \le n \le 10^3$。

L & M. Yesterday Once More 题解

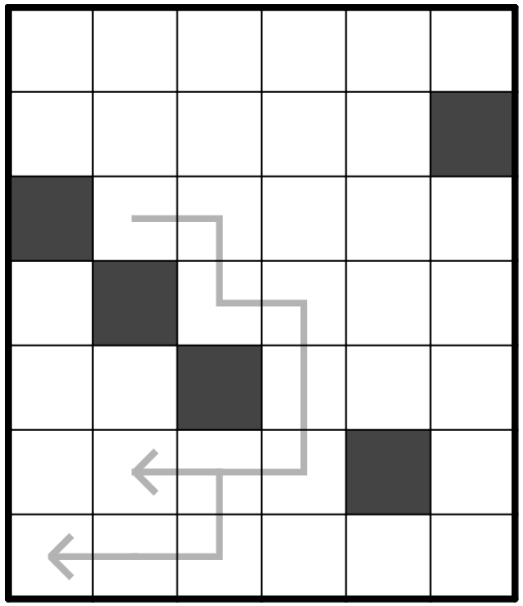
- 先考虑能不能从最左边溜下去。
- 构造 $\texttt{D} \cdot n + \texttt{RDDL} + \texttt{D} \cdot n$，那么只会在下面两种情况出问题：





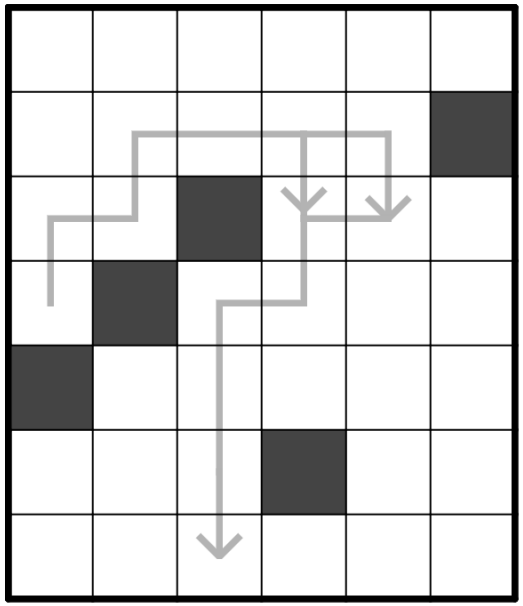
L & M. Yesterday Once More 题解（续）

- 考虑怎么处理第一种情况。
- 构造 RDDLL $\cdot n$，那么它可以一直沿着障碍往下走，直到溜进空里：



L & M. Yesterday Once More 题解（续）

- 考虑怎么处理第二种情况。
- 构造 $\texttt{URRD} \cdot n + \texttt{L} \cdot n + \texttt{DL} + \texttt{D} \cdot n$，那么它可以一直沿着障碍往上走，直到溜进空里，再原路返回并往下走到第 $n+1$ 行：

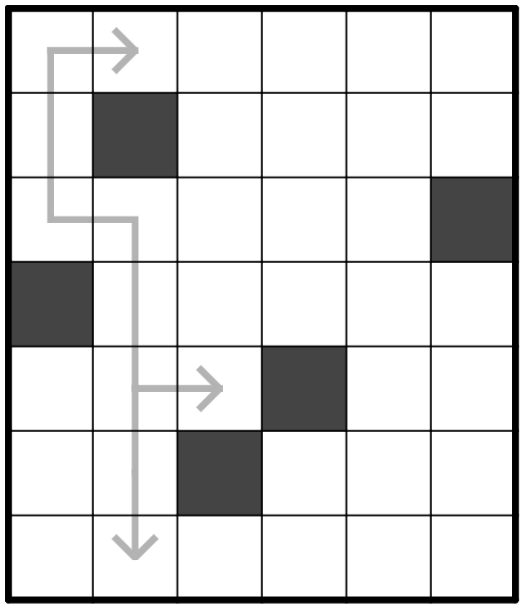


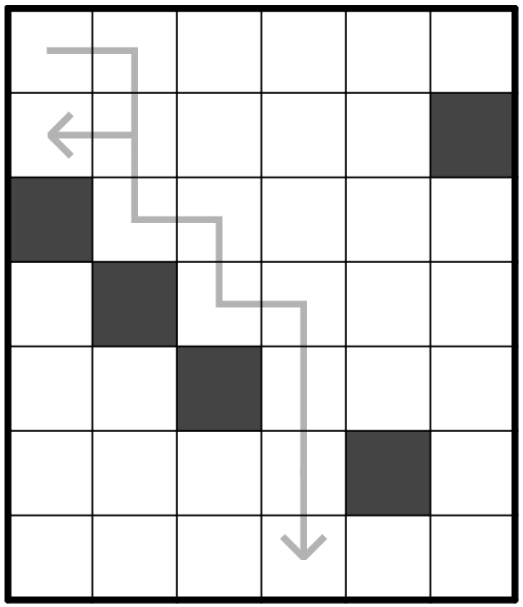
L & M. Yesterday Once More 题解（续）

- 至此总移动次数做到了 $13n + O(1)$，可以通过 Easy Version，接下来考虑进一步优化。

L & M. Yesterday Once More 题解（续）

- 注意到，初始时往下溜的 $2n$ 次移动和处理第一种情况的 $5n$ 次移动可以直接合并成 $\texttt{RDLD} \cdot n$：





L & M. Yesterday Once More 题解（续）

- 同时，后 $6n$ 步中的 URRD 可以只循环 $n-1$ 次。
- 将两部分合并，可以得到最终答案为
  $\texttt{RDLD} \cdot n + \texttt{URRD} \cdot (n-1) + \texttt{L} \cdot n + \texttt{DL} + \texttt{D} \cdot n$。
- 总移动次数做到了 $10n-2$，可以通过 Hard Version。

Thank you!