2025 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

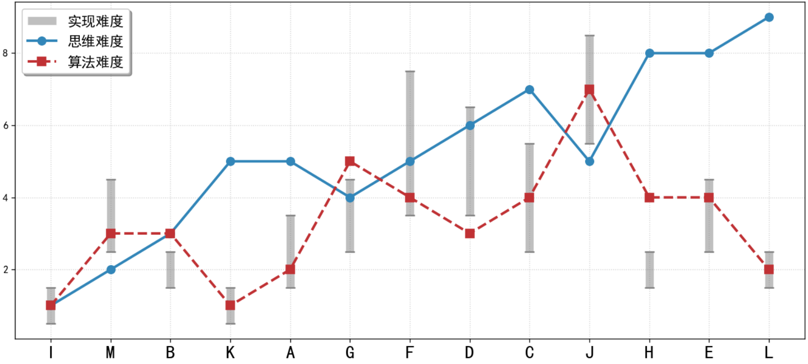
2025 年 11 月 16 日

概况

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

概况 (cont’d)

- 实现难度：代码量和细节讨论，柱状区间体现不同思考深度、解法带来的实现难度差异
- 思维难度：临场的观察建模
- 算法难度：赛前的知识储备



A. Square Kingdom

题意

- 给一个长为 n 的数列 $h_i = \left(i + \frac{b}{a}\right)^2$ $(i = 1, 2, \ldots, n)$，求 $h_j - h_i$ $(1 \le i < j \le n)$ 从小到大排序后的第 k 个元素，以最简分数的形式输出。
- $2 \le n \le 10^{12}$, $1 \le k \le 10^{12}$, $1 \le a \le 10^6$, $0 \le b \le 10^{12}$。

- 赛中通过：36 支队伍，封榜前 23 支队伍。
- 首次通过：北京大学-不渡轮回（55 分钟）。

A. Square Kingdom (cont'd)

- 首先有 $h_j - h_i = \left(j + \frac{b}{a}\right)^2 - \left(i + \frac{b}{a}\right)^2 = (j-i)(i+j+\frac{2b}{a})$。
- 那么 $h_j - h_i$ 都是 $\frac{1}{a}$ 的整数倍，二分答案 $\frac{res}{a}$，要判断是否有 $\geq k$ 组 (i,j) 满足 $(j-i)(i+j+\frac{2b}{a}) \leq \frac{res}{a}$。
- 枚举 $j-i=d$，可得 $i \leq \frac{res-2bd-ad^2}{2ad}$，另有 $1 \leq i \leq n-d$，对合法 i 的数量求和即可。
- 这里 d 只需要枚举到 $\sqrt{2k}$ 左右，这是因为如果有 $h_j - h_i \leq \frac{res}{a}$，那么所有 $i \leq i' < j' \leq j$ 都有 $h_{j'} - h_{i'} \leq \frac{res}{a}$。
- 因此有 $res \leq a(h_{\sqrt{2k}+1} - h_1) = a(2k+\sqrt{2k}) + 2b\sqrt{2k}$，不超过 5×10^{18}，实际上可以参考第四组样例设置二分上界。
- 二分的时候有加法溢出的风险，最后输出答案也要约分。

B. Buggy Painting Software I

题意

- 给定 $n \times m$ 的目标图像的像素矩阵，可以免费创建图层、花费 a 使用单次画笔、花费 b 使用单次橡皮，求使得合成图像匹配目标图像的最小总代价。
- $1 \leq n, m \leq 500$, $1 \leq a, b \leq 10^6$。

- 赛中通过：371 支队伍，封榜前 352 支队伍。
- 首次通过：北京大学-柚子超玩会（12 分钟）。
- 正式队伍首次通过：北京大学-不渡轮回（20 分钟）。

- 使用画笔工具产生的像素一定会显示在最终图像中，否则浪费代价。
- 因此最终在顶层创建空图层一次性将需要用画笔工具产生的像素画完一定不劣。
- 底层只使用橡皮工具，那么一定是初始色互不相同的纯色图层按最优顺序排，为了显示某个颜色的像素从上往下擦到这个颜色为止。
- 要显示出从上到下第 k 个纯色图层的像素，需要使用 $k-1$ 次橡皮工具。显然同种颜色的像素要么全用画笔工具，要么全用橡皮工具。

B. Buggy Painting Software I (cont'd)

- 设矩阵中可见值 i 出现了 c_i 次，设 nm 种颜色中出现次数第 i 大的是 r_i，即 $c_{r_1} \geq c_{r_2} \geq \cdots \geq c_{r_{nm}}$。
- 设创建 k 个纯色图层，它们的颜色出现次数排名分别是 $e_1 < e_2 < \cdots < e_k$，贪心地将颜色 r_{e_i} 放在第从上到下第 i 层，总代价是：

$$\left(a \sum_{i \notin [e_1, e_2, \ldots, e_k]} c_{r_i}\right) + \left(b \sum_{i=1}^{k} \left(c_0 + (i-1) c_{r_{e_i}}\right)\right)$$

B. Buggy Painting Software I (cont'd)

- 上式满足 $a \cdot c_{r_{e_k}} \geq c_0 + b(k-1)c_{r_{e_k}}$，否则将颜色 r_{e_k} 改为用画笔工具不劣。
- 可以放缩到 $a \geq b(k-1)$，这说明最优情况下应贪心地令 $e_i = i$，让出现次数第 $1, 2, \ldots, k$ 大以此产生 $0, b, 2b, \ldots, (k-1)b$ 倍的贡献，让其他出现次数更小的颜色产生 a 倍贡献。
- 从 0 到 nm 枚举 k，计算下式的最小值即可：

$$\left(a\sum_{k+1}^{nm} c_{r_i}\right) + \left(b\sum_{i=1}^{k} (c_0 + (i-1)c_{r_i})\right)$$

- 瓶颈在于排序，时间复杂度为 $O(nm\log(nm))$。

C. Buggy Painting Software II

题意

- 通信题。第一次运行需要构造 n 个长度为 $3m$ 的序列，每个序列中 1 到 m 的整数恰好出现三次，要求第 i 个序列加密了 x_i 这个数（x_i 的范围也是 1 到 m）。然后 SPJ 会独立地将每个序列用某个二划分 (S_0, S_1) 映射成二进制序列，第二次运行需要从这些二进制序列中解密出每个 x_i。
- $2 \leq n, m \leq 5 \times 10^5$，$n \times m \leq 10^6$。

- 赛中通过：6 支队伍，封榜前 2 支队伍。
- 首次通过：西安交通大学-六兆年零一夜的 ACM（113 分钟）。

- 本题做法很多。最自然的想法是将一个 $1, 2, \ldots, m$ 放在每个序列的开头，从而解密的时候能够直接知道 (S_0, S_1)。
- 剩下的位置希望尽可能差别大一些，不然 SPJ 容易构造出 (S_0, S_1) 攻击使得映射后无法区分。
- 所以加密 x_i 时尝试将 $1, 2, \ldots, m$ 循环移位 x_i 位放在开头的 $1, 2, \ldots, m$ 后面。由裴蜀定理知，当 m 为质数时，$x_i \neq x_j$ 不会被映射到同个二进制序列，否则 $\gcd(|x_i - x_j|, m) = 1$ 可以推出二进制序列要么全 0 要么全 1，即反证。
- 上述做法在 m 为合数时仍可被攻击。幸运的是，剩余位置有 $2m$ 个而非 m 个，这启发我们耗费更多位置来构造等价于质数的加密方式。

C. Buggy Painting Software II (cont'd)

- 根据伯特兰-切比雪夫定理，m 到 $2m$ 之间必然存在一个质数 p。
- 从 $1, 2, \ldots, m, 1, 2, \ldots, m, 1, 2, \ldots, m$ 结构出发，前 m 个位置仍然用于得到 (S_0, S_1)，接下来 p 个位置 $1, 2, \ldots, m, [1, 2, \ldots, m, 1, 2, \ldots, p-m], p-m+1, \ldots, m$ 用于循环移位 x_i 位。
- 解密时，用 (S_0, S_1) 推出 $1, 2, \ldots, m, 1, 2, \ldots, p-m$ 循环移位前的二进制序列，要知道循环移位了几位是一个经典问题，可以通过倍长后 KMP 算法解决，也可以算序列中值为 1 的下标总和在模 p 意义下除以 1 的个数。
- 单次加密、解密的复杂度是 $O(m)$，总时间复杂度为 $O(nm)$。

D. LED Display Renovation

题意

- 给一个有 n 个数字块的标准七段 LED 显示屏，给出每个数字段的初始状态（正常、常亮、常灭），可以将至多 k 个数字段翻新正常状态，问最多能正确展示多少个整数以及得到最大值的方案数。
- $1 \leq n \leq 9$, $0 < k < 7n$。有 $T \leq 100$ 组数据。

- 赛中通过：10 支队伍，封榜前 7 支队伍。
- 首次通过：浙江大学-乘着西风出发喽！（121 分钟）。

D. LED Display Renovation (cont'd)

- 记 $f_{i,j,0/1}$ 表示最高 i 位翻新了 j 个数字段，是否允许前 i 位全空的情况下，最多能正确展示多少个整数。
- 转移枚举下一位的 2^7 种翻新方案，模拟一下求出每种翻新方案可以让这一位展示哪些数字，注意区分 0 和其他数字，以及展示空的情况。
- 对于方案数，需要同时维护取到最大值的方案数以及所有的方案数，对于某一位翻新之后只能展示 0 种数字的转移，要将所有的方案数转移给取到最大值的方案数。

E. Play It by Ear

题意

- 有 $2n$ 张标号 1 到 $2n$ 的牌和 m 个按顺序完成的任务，第 i 个任务需要在完成第 $i-1$ 个任务之后打出一次标号 a_i 的牌。初始牌堆在 $(2n)!$ 种排列中等概率随机，摸牌堆顶 n 张牌作为起始手牌，牌堆其余牌顺序不可见，每轮出牌时任选手牌中一张牌置于牌堆底部并从牌堆顶摸一张。求最优纯策略下完成所有任务轮数的最小期望值，答案对 $998\,244\,353$ 取模。
- $2 \le n \le 5\,000$，$1 \le m \le 2 \times 10^5$。

- 赛中通过：2 支队伍，封榜前 1 支队伍。
- 首次通过：北京大学-不渡轮回（201 分钟）。

E. Play It by Ear (cont'd)

- 对于排列 π，我们容易求出理论轮数下界。设 f_i 为该排列下完成前 i 个任务的最小轮数，则 $f_i = 1 + \max\{f_{i-1}, f_j + n\}$，其中 j 为上一个目标牌标号和 a_i 相同的任务序号，特别地，对于目标牌标号是任务里第一次出现的 i，须将 $f_j + n$ 这一项换成 $\max\{pos_{a_i} - n, 0\}$（pos_{a_i} 表示 a_i 在 π 中的出现位置）。
- 事实上，该下界一定能通过以下纯策略实现：如果手牌里有当前任务的牌，立即打出，否则打出离在任务序列里下一次出现最远的那张牌。由于后者发生时，根据鸽巢原理最远的牌一定距离大于等于 n，所以现在打出一定能来得及在任务真正需要它的时候回到手中。

E. Play It by Ear (cont'd)

- 考虑期望到概率的转化：

$$E[f_m]=\sum_{k=1}^{\infty}kP(f_m=k)=\sum_{k=1}^{\infty}P(f_m\geq k)=\sum_{k=0}^{\infty}(1-P(f_m\leq k))$$

- 令 $P(f_m\leq k)$ 取大于 0 小于 1 的 k 的范围是 $O(n)$ 长的连续区间，对于这样的 k，可以从最后的 $f_m\leq k$ 往前倒推出每个 f_i 的上界约束是多少（例如 $f_i=1+\max\{f_{i-1},f_j+n\}\leq u$ 可拆成 $f_{i-1}\leq u-1$ 和 $f_j\leq u-n-1$，约束之间需取 $\min$），最后倒推出该 k 下 π 中每个值出现位置 $pos_1,pos_2,\ldots,pos_{2n}$ 分别的上界约束，排序后优先填约束紧的数，乘法原理即可，时间复杂度为 $O(nm\log n)$。
- 实际只需对单个 k 求解 $pos_1,pos_2,\ldots,pos_{2n}$ 的上界约束，k 变化时这些约束要么不变要么变化相同的差值，时间复杂度降至 $O(m+n\log n)$。

F. The Bond Beyond Time

题意

- 给一个 n 个点 m 条边的简单连通无向图 G，要给每条边定向，现在有两个人分别在点 x 和 y $(x \neq y)$，每一步两个人会各自同时沿着某条出边走到下一个点，如果没有出边就留在原地，构造一个定向使得两个人不可能走到同一个点。
- $2 \leq m \leq 300, 1 \leq m \leq \frac{n(n-1)}{2}$。

- 赛中通过：52 支队伍，封榜前 33 支队伍。
- 首次通过：北京大学-为你而来（73 分钟）。

F. The Bond Beyond Time (cont'd)

- 本题建议 Think twice, code once，优秀的实现仅需要两层 BFS，如果涉及 LCA 或 Tarjan 很容易写麻烦。
- 如果 x 和 y 在 G 上不相邻，那么所有端点有 x 或 y 的边都指向 x 或 y，这样通过将两个人困在原地满足要求。
- 如果 x 和 y 在 G 上相邻，因为 $x \to y$ 或 $y \to x$ 之一存在，不论如何定向必须让终点的人也移动，同理下一步两人仍然必须同时移动，为了永不相遇，只能将两人引诱到一个环里，
- 此时若 G 是一棵树则无解，否则一定有解，接下来我们默认 x 和 y 在 G 上相邻且 G 有环。

F. The Bond Beyond Time (cont'd)

- 首先，环不能有弦，否则弦无论如何定向都可能在两人绕环的过程中被经过导致相遇，幸运的是，总能通过该弦分割出更小的环。
- 所以我们找包含 $x \leftrightarrow y$ 的最小环，其一定没有弦，最理想的实现是断掉这条边然后跑 x 到 y 的 BFS 最短路。
- 如果 $x \leftrightarrow y$ 不在任何环中（是 G 的割边），需要将两人引诱到外部的环中。
- 从 x, y 开始 BFS，每遇到一条边就找包含这条边的最小环，找到了就将环定向成有向环并将其他边指向该环。
- 由于 G 最多只有 $O(n)$ 条割边，内层 BFS 最多被调用 $O(n)$ 次，而断边找最小环是 $O(n^2)$，因此总时间复杂度为 $O(n^3)$。实际上有 $O(n+m)$ 的解法，但是 SPJ 已经是 $O(n^3)$ 的。

G. Collision Damage

题意

- 给定凸多边形 P 和 Q，在 $area(P \cap (Q + \mathbf{t})) > 0$ 的前提下均匀随机选择 $\mathbf{t}$，求 $area(P \cap (Q + \mathbf{t}))$ 的期望。
- $3 \leq n, m \leq 1\,000$。

- 赛中通过：34 支队伍，封榜前 21 支队伍。
- 首次通过：复旦大学-做 AC 梦（94 分钟）。

G. Collision Damage (cont'd)

- 根据期望线性可加，考虑一个面元 $x \in P \cap (Q + \mathfrak{t})$ 的概率，首先有 $x \in P$，然后要选取 $\mathfrak{t}$ 也就是平移 Q 使得 x 在平移后的 Q 内，其概率为 $\frac{area(Q)}{area(D)}$，因此期望就是 $\frac{area(P) \cdot area(Q)}{area(D)}$。通过交换积分序也可以得到同样的结论。
- 根据上述分析也可以知道 $D = P - Q$，也就是 P 和 $-Q$ 的闵可夫斯基和，可以 $O(n + m)$ 求解。
- 由于数据范围较小，也可以枚举 P 的顶点和 Q 的顶点两两做差之后对 $O(nm)$ 个点求凸包，复杂度是 $O(nm \log(nm))$。

H. Cute Young Diagram Counting

题意

- 对长度为 n 的序列 $a_1, a_2, \ldots, a_n$ 的每个前缀求出该杨图能通过若干次操作得到的不同杨图个数，操作是将任意一个子图共轭并要求全局仍然是个杨图，答案对 $998\,244\,353$ 取模。
- $1 \le n \le 10^6$。

- 赛中通过：2 支队伍，封榜前 2 支队伍。
- 首次通过：北京大学-不渡轮回（162 分钟）。

H. Cute Young Diagram Counting (cont'd)

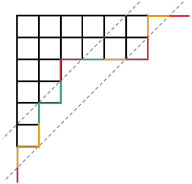
- 操作能得到每条反对角线上格子数保持不变的所有杨图。
- 必要性：操作永远只会改变反对角线上格子之间的位置，不会改变数量。
- 充分性：可以通过依次贪心地操作尽可能提高第一行、第二行、……的长度来得到 $\lambda_1 > \lambda_2 > \cdots > \lambda_m$ 的杨图，容易证明该杨图对于“每条反对角线上格子数不变”下是唯一的。又因为操作可逆，反对角线上格子数相同的杨图互相可达。

充分性证明

最大化 λ_1，即找到所在反对角线最靠右下的格子并操作该格子所在列的第一行的格子，由于该列的长度会缩短为操作前的更小的 λ_1，整体仍是一个杨图，此时不再考虑第一行（可以视作删去）便得到了第 $1, 2, \ldots, \lambda_1$ 条反对角线上的格子数同时减少 1 的子问题，归纳上述正确性即证。

H. Cute Young Diagram Counting (cont'd)

- 接下来做法很多，最快捷的是构建杨图和其轮廓线的双射并计数轮廓线（假设左边界向下无限延伸、上边界向右无限延伸）：
- 设 t_i 表示第 i 条反对角线上的格子数，令差分序列 $d_i = t_{i+1} - t_i$，显然轮廓线会恰好经过 $d_i + 1$ 个该对角线上格子的下侧、$d_i + 1$ 个该对角线上格子的右侧，不妨考虑轮廓线在第 $i - 1$ 条对角线和第 $i + 1$ 条对角线之间的部分，如图所示：



- 其中绿色段为反对角线 $i - 1$ 上 $d_{i-1} + 1$ 个格子的下侧、$d_{i-1} + 1$ 个格子的右侧，黄色段为反对角线 i 上 $d_i + 1$ 个格子的下侧、$d_i + 1$ 个格子的右侧。

H. Cute Young Diagram Counting (cont'd)

- 轮廓线在该区域的两端必然为黄色段（因为两侧均从无限远穿过第 $i+1$ 条对角线），而由于其他段依次必然为同色的下侧、右侧捆绑出现，实质上轮廓线在该部分的不同情况数等于 $d_{i-1}+1$ 个绿色段、d_i 个黄色段的组合数，即 $\binom{d_{i-1}+d_i+1}{d_i}$。
- 又由于不同 i 的区域分别独立地选择一种情况总能拼成一条完整的轮廓线，乘法原理可得总情况数：

$$\prod \binom{d_{i-1}+d_i+1}{d_i}$$

- 回到原问题，从左到右加入 $a_1, a_2, \ldots, a_n$，显然每加入一个 a_i 只会影响差分序列 d 的两个位置和最多四个组合数，分别重新计算乘法项的贡献即可，时间复杂度为 $O(n)$。

题意

- 按照时间顺序给出一场比赛的 n 个 AC 提交，按照如下规则判断每个提交是否会发气球，并输出气球对应的题号：
  - 封榜前（< 240 分钟）每个题的首次 AC 提交都发气球；
  - 封榜后（≥ 240 分钟）每个题的首次 AC 提交，如果已经获得的气球 < 3 就发气球。
- $1 \leq n \leq 5\,000$。

- 赛中通过：410 支队伍，封榜前 409 支队伍。
- 首次通过：东北育才学校-东北大学附属中学（3 分钟）。
- 正式队伍首次通过：北京大学-不渡轮回（3 分钟）。

I. Volunteer Simulator (cont'd)

- 按照题意模拟，本场比赛的气球也是按照这个规则发的。

J. The Echoes of Chronos

题意

- 给定一个整数序列 $a_1, a_2, \ldots, a_n$，每次操作可以将序列里所有 x 同时变成 $x+1$ 或者 $(x-1) \bmod m$，有 q 个询问 (l, r, v)，求使得 $a_l = a_{l+1} = \ldots = a_r = v$ 的最小操作次数。
- $n, q \le 2 \times 10^5$，$m \le 10^9$。

- 赛中通过：3 支队伍，封榜前 3 支队伍。
- 首次通过：北京大学-柚子超玩会（199 分钟）。
- 正式队伍首次通过：中山大学-破晓之旅（205 分钟）。

J. The Echoes of Chronos (cont'd)

- 考虑单个询问 (l, r, v)，长为 m 的环被 $a_l, a_{l+1}, \ldots, a_r, v$ 分割成若干段，最小操作次数就是 m 减去最长段的长度。
- 选取一个整数 $B = O(\sqrt{n})$，将询问按 $\lceil \frac{l}{B} \rceil$ 分组，组内按照 r 从大到小排序。
- 对原序列离散化，构建一个循环的双向链表顺序维护环上的分割点，这样可以快速支持移除一个分割点，然后通过记录本次修改的数据来实现回滚移除一个分割点的过程。
- 对于第 1 组里的询问 (l, r, v)，移除所有下标 $> r$ 的分割点，然后移除所有下标 $< l$ 的分割点，处理询问，再回滚掉移除所有 $< l$ 的分割点的过程。
- 处理完第 k 组询问之后，移除所有下标 $\leq kB$ 的分割点，再继续处理下一组询问。

J. The Echoes of Chronos (cont'd)

- 处理询问还需要插入分割点 v，但是插入 v 最多影响一个段。
- 考虑维护最长两个段的长度以及对应是哪个段，如果 v 不插在最长段内，直接取最长段的长度即可，否则需要考虑最长段被 v 分割之后的两个段的长度以及次长段的长度。
- 移除一个分割点会合并分割点两侧的段，这样需要删除两个段，如果删除的两个段正好是维护的最长两个段，说明合并之后最长段的长度至少是合并前次长段的两倍。
- 这样在合并后即使向最长段加入分割点，也不需要再考虑这次合并之前维护的次长段。
- 复杂度 $O((n+q)\sqrt{n})$。

K. Relay Jump

题意

- 有 n 只平面上的青蛙，每只青蛙被刺激时会跳到对称于另一只青蛙的位置并刺激这只青蛙，从编号为 s 的青蛙开始，直到某只编号为 t 的青蛙被刺激时选择停下为止。你不知道跳跃的完整过程，需要通过 s、每只青蛙的初始和最终位置来确定 t 的值。
- $2 \leq n \leq 10^5$。

- 赛中通过：140 支队伍，封榜前 108 支队伍。
- 首次通过：北京大学-不渡轮回（27 分钟）。

K. Relay Jump (cont'd)

- 跳跃的完整过程非常复杂且难以推测，这提示我们寻找跳跃前后的不变量。
- 该不变量满足：当前被刺激的青蛙对不变量产生独特的贡献，其他青蛙产生相同的贡献，这样方便锁定被刺激的青蛙的编号。因此设 $\Phi_P(i)$ 表示被刺激的青蛙为 i 时用位置集合 P 算出的不变量（类似势能）。
- 当青蛙 i 跳过青蛙 j 时，位移向量 $\overrightarrow{P_iP_i'}$ 等于被刺激位置转移向量 $\overrightarrow{P_iP_j'}$ 的两倍，可以写成 $P_i' - P_i = 2P_j - 2P_i$（x, y 两个分量分别做标量运算），移项即 $2P_i - P_i = 2P_j' - P_i'$。
- 这个等式是推出不变量的关键，在不变量等式 $\Phi_P(i) = \Phi_{P'}(j)$ 两边加上该等式得：

$$\Phi_P(i) + 2P_i - P_i = \Phi_{P'}(j) + 2P_j' - P_i'$$

K. Relay Jump (cont'd)

- 为了让 j 不论选择谁等式都成立，可以解出：

$$\Phi_P(i) = \left(\sum_{k=1}^{n} P_k\right) - 2P_i$$

- 则：

$$\left(\sum_{k=1}^{n} P_k\right) - 2P_s = \left(\sum_{k=1}^{n} Q_k\right) - 2Q_t$$

- 由于 $Q_1, Q_2, \ldots, Q_n$ 互不相同，t 可以被唯一解出。

L. Leo

题意

- 实现一个三色逻辑计算机，满足 n 个输入端从 $\{\mathsf{R}, \mathsf{G}, \mathsf{B}, *\}$ 中取时，计算机总能算出从左到右遇到的第三种不同颜色。要求至多用 $6n$ 个元件（详见完整题意）。
- $3 \leq n \leq 10^5$。

- 赛中通过：0 支队伍。

L. Leo (cont'd)

- 第一步：消除大部分 $*$，只允许 $*$ 出现在一小段后缀中。
- 其动机是，AND 容易受到 $*$ 干扰导致始终产生 $*$ 而非预期之内的结果，于是我们希望每个 $*$ 都能被就近的颜色取代。
- 不妨考虑让每个 $*$ 被其后最近的颜色取代（如果存在的话），这样做的好处是，输入端从左到右遇到的第一种颜色就在第一个位置，只要能设法找到从左到右首个异于该颜色的位置，OR 一下就是答案。
- 不难发现 x OR y OR y 是一个“二路优先选择门”，该元件可以在 x 非 $*$ 仍然返回 x，而 x 为 $*$ 时返回 y。
- 所以我们可以使用该元件倒着扫描输入端，通过“后缀优先选择”达成目的，我们设此时的序列为 $a_1, a_2, \ldots, a_n$：

	$*$	R	R	$*$	B	G	$*$	R	$*$
a	R	R	R	B	B	G	R	R	$*$

L. Leo (cont'd)

- 第二步：标识出从左到右遇到的第二种不同颜色的位置。
- 延续第一步的思路，碍于黑盒机制我们不可能直接点对点地 OR 到第二种颜色，因为它可能在任何位置，我们应当侧重如何让它在任何位置时都能被标识，具体如下：

	*	R	R	*	B	G	*	R	*
a	R	R	R	B	B	G	R	R	*
f	R	R	R	*	*	*	*	*	*
b	R	R	R	G	B	G	R	R	*

- 令 $f_i = f_{i-1}$ AND a_i 和 $b_i = f_{i-1}$ OR a_i，显然 b 序列仅在从左到右遇到的第二种不同颜色的位置处和 a 序列不同。

- 第三步：计算出从左到右遇到的第三种不同颜色。
- 从 a, b 序列只有一个位置不同出发，容易想到构造类似 $(a_1 - b_1) + (a_2 - b_2) + \cdots + (a_n - b_n)$ 的代数运算来抵消相同位置，而差异位置贡献出目标颜色。
- 而 OR 刚好具备优秀的代数性质，分别记从左到右遇到的第一、二、三种颜色为 $0, 1, 2$，则 x OR $y \equiv -x - y \pmod 3$，进而偶数项的 OR 链：
x_1 OR x_2 OR $\cdots \equiv x_1 - x_2 + x_3 - x_4 + \cdots \pmod 3$。
- 也即，a_1 OR b_1 OR a_2 OR b_2 OR $\cdots$ OR a_n OR b_n 的代数结果在 mod 3 意义下就是 $a_k - b_k \equiv 1 - 2 \equiv 2$（$k$ 是两序列的差异位置），刚好是第三种颜色，并且即使 a, b 有等长的 $*$ 后缀，该颜色结果显然会保留到最后。
- 三个步骤有机结合，成功构造了 $6n - O(1)$ 个元件的计算机。

M. The End?

题意

- 8 支队伍打单败淘汰赛，要给每支队伍排 1 到 8 的种子，分别给出每支队伍在比赛时作为种子编号较小的队伍以及作为种子编号较大的队伍的实力值，实力值 x 对上实力值 y 的队伍的胜率是 $\frac{x}{x+y}$，最大化队伍 1 的胜率。

- 赛中通过：383 支队伍，封榜前 367 支队伍。
- 首次通过：北京大学-不渡轮回（15 分钟）。

- 枚举队伍种子的全排列。
- 由于单败淘汰赛程是一棵树，在赛程的每个节点计算队伍胜率的分布列即可。

Thank you!