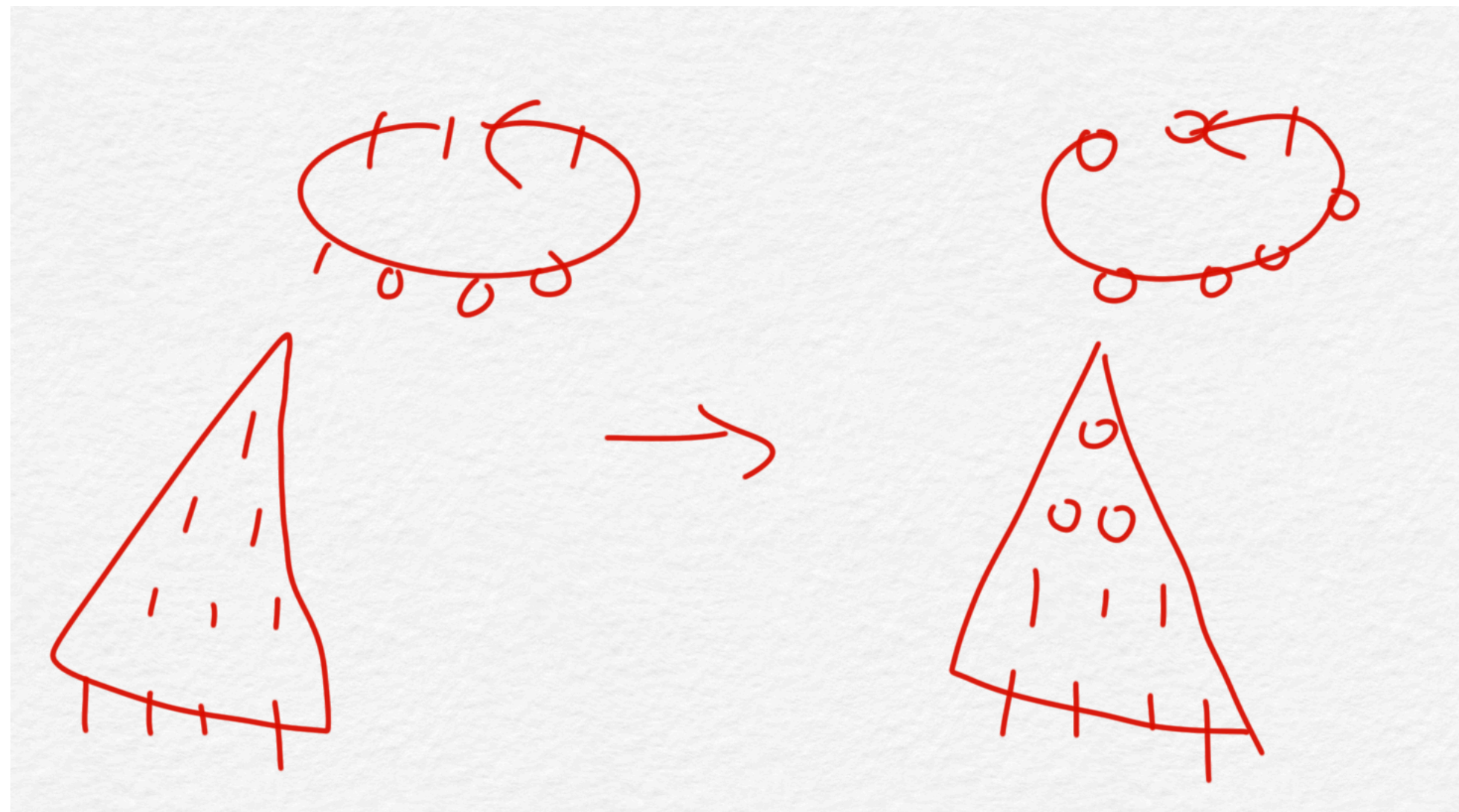


EC Final 2022 简要讲解

EC Final 2022 裁判组 2023.3.26

A - Coloring

- 整个图的结构是基环树。
- 考虑变换，一定是环上连续一段0一段1变换，然后随着环上01变换，子树里也会变成01分层的情况。



A - Coloring

- 子树里面如果 01 分成了最多 k 次，那么环上至少 01 变换 k 次。
- 首先考虑子树 dp，记录一下每个子树里面 01 变换的最多次数。
- 然后考虑环上，我们可以枚举 1 在哪一段，枚举环上 01 变换的次数。
- 最后把枚举 1 在哪一段这部分，用前缀和或者 dp 优化即可。
- 时间复杂度 $O(n^2)$ 。

B - Binary String

- 考虑一段长度大于 1 的连续的 0:
- ...0101000000101... 会变成 ...1010000001010...
- 相当于这段 0 整体向左移动了一位; 对于 1 连续段, 则为整体向右移动了一位。
- 如果 1 连续段和 0 连续段相邻:
- ...0111100001... 会变成 ...1011100010...
- 相当于两个连续段相撞, 各自减少 1 的长度。

B - Binary String

- 会不断执行直到没有 0 连续段或 1 连续段。
- 不妨设 1 段总长度不小于 0 段总长度，找到一个起点位置，使得每个前缀中 1 段长度和均不小于 0 段长度和。
- 那么就可以求出每个 0 连续段会在什么时候消失，并求出所有 0 连续段消失后的 01 序列。最后求一次结尾串的最小整周期长度即可。
- 时间复杂度 $O(n)$ 。

C - Best Carry Player 2

- 考虑从高位往低位数位dp, $dp[i][j][k]$, 表示做到第 i 位, 一共进了 j 位, 后面是否有向前面进位, 每次考虑下一位是否进位, 以及进位/不进位情况下填的最小数字即可。
- 时间复杂度 $O(\log^2 n)$ 。
- 注意一些 corner case:
 - 数字必须是正数, 当 $k = 0$ 的时候需要简单特判。
 - 答案可能超过 10^{18} , 比如 $n=10^{17}$, $k=18$, 这个时候可以去除末尾所有 0 再做, 或者用 int128。

D - Minimum Suffix

- 首先考虑原问题，也就是给定一个字符串，问每个前缀最小后缀如何解决。
- Lyndon 分解：把一个字符串分解成若干个 Lyndon 串，也就是每个都小于它所有循环同构的串，并且每个都大于等于后一个串。
- 这个问题可以用 Lyndon 分解理论，使用 Duval 算法在线性时间复杂度内解决。

D - Minimum Suffix

- 考虑现在的问题，给定了原问题的答案，可以根据 Duval 算法，把现在串表示成若干个 Lyndon 串，每个 Lyndon 串内部，会有一些相等和小于关系。
- 比如 $p=[1, 2, 1, 4, 1]$ ，可以得到 $s[1] = s[2] < s[3]$, $s[1] = s[4] < s[5]$ 。
- 每个 Lyndon 串之间，会有前面串大于等于后面串的关系。
- 要求字典序最小，每个串内部可以从前往后贪心，串与串之间，从后往前贪心，每次都找到一个字典序大于等于后面串的最小串。
- 因为 Lyndon 串的结构，所有的小于关系都是从前往后的，所以也可以逐位贪心，确定每个串的最小串。
- 时间复杂度 $O(n)$ 。

E - Map

- 可以证明，最优的路径一定是先从大到小传送若干次，然后移动，然后从小到大传送若干次。
- 如果在中间过程中移动，那么这段路径放到最后去移动一定更优。
- 问题可以看成 s 从大到小传送若干次到达 s' ， t 从大到小传送若干次 t' ，然后从 s' 走到 t' 即可。
- 因为次数限制很小，所以可以 $O(n^2)$ 枚举。
- 关于如何求这个相似变换，最简单的方法是复数法，一个相似变换一定可以表示成 $f(z) = Az + b$ ，或者 $A\bar{z} + b$ 。

F - Inversion

- 记 $f(l, r)$ 表示 l 到 r 的答案。
- 可以发现 $[a_l > a_r] = f(l, r) - f(l + 1, r) - f(l, r - 1) + f(l + 1, r - 1)$ 。
- 直接排序，用上面方法比较，可以做到 $4n \log n$ 的时间复杂度。
- 考虑插入排序，每次二分每个数在什么位置，插入一个数 $p[i]$ 的时候， $p[1]$ 到 $p[i - 1]$ 都已经确定，所以 $i - 1$ 的答案可以直接计算出来不需要询问，所以只需要两次查询。
- 查询次数 $2n \log n$ ，时间复杂度 $O(n^2)$ 。

G - Rectangle

- 根据直线的情况分类，有三条横线，一横两竖，以及对称情况。
- 对于三条横线，只和所有区间的 y 坐标相关，所以是一个一维问题。
- 考虑枚举中间线的位置，上面和下面的线是上面未被覆盖和下面未被覆盖的区间的交。

G - Rectangle

- 对于一横两竖，考虑对横线做扫描线，变成了你要支持删除一个区间，插入一个区间，问用两条点覆盖剩下区间的方案数。
- 假设所有区间为 $[l_i, r_i]$ 。首先左边的点 $L \leq \min(r_i)$ ，右边的点 $R \geq \max(l_i)$ 。
- 维护 $f(L)$ 表示左边的点在 L 位置，右边的点 R 的上界。每次插入一条线段 $[l, r]$ ，会有如果 $L < l$ ，那么 R 一定要 $\leq r$ 。也就是对于一个前缀，有一个 $\leq r$ 的限制。因为限制需要删除，所以不能直接覆盖。
- 可以考虑线段是直接维护这些限制的后缀最小值和它们的和。这个东西可以用线段树维护，常被称为楼房重建的技巧，时间复杂度是 $O(n \log^2 n)$ 的。

H - Chinese Checker

- 枚举哪个棋子作为起点，然后根据题意 BFS 即可。

I - Chase Game

- 首先注意到，如果两个人到达同一个点，那么一定是按照最短路直接走到终点。
- 也就是只有一开始一段 Pang 没有动的需要考虑，Pang 传送之后可以直接计算答案。
- 所以有效的状态是 $O(n)$ 的，预处理 Pang 和 n 号点到所有点的最短路之后，使用 Dijkstra 计算即可。
- 时间复杂度 $O(n \log n)$ 。

J - Chase Game 2

- 首先对于每个叶子一定需要加边，否则如果起始点在这个点和它的父亲，直接无解了。
- 然后这个叶子不能只和它的兄弟加边，否则上面的情况也无解。
- 所以如果是一个 star，那么一定无解。
- 否则考虑叶子总数和最大值 sum, mx ，答案一定不会小于 $\max((sum + 1)/2, mx)$ 。
- 不难证明，这个下界可以取到。

K - Magic

- 换一种方式考虑：
- 要选择若干个位置，使得这些位置最后贡献给答案，要求合法且选择的位置数量最多。
- 对于区间 $[l_1, r_1]$ 和 $[l_2, r_2]$ ，若 $l_1 < l_2 < r_1 < r_2$ ，那么如果选择 l_2 ，就意味着 $[l_1, r_1]$ 要在 $[l_2, r_2]$ 之前执行；同理如果选择 r_1 ，就意味着 $[l_2, r_2]$ 要在 $[l_1, r_1]$ 之前执行。
- 所以 r_1 与 l_2 不能同时选择，所以一定是一个建无向图后的独立集。
- 下面证明这是充分的。

K - Magic

- 只要选择独立集后不会成环，那么就足够说明这是合法的。
- 如果成环，假设环上选了一个区间 $[l, r]$ 的右端点 r ，导致它比某个区间 $[l', r']$ 要晚执行 ($l < l' < r < r'$)。
- 由于 l' 不能被选择，那么下一个环上的点一定是由于选择了 r' 导致的。
- 那么以此类推，在这条链上的区间 r 一定单调递增，所以不可能成环。
- 因此这个限制是充分的。

K - Magic

- 所以问题转化为了最大独立集问题。
- 又由于只有左端点和右端点之间的边，这是一个二分图最大独立集问题。
 - bitset 优化匈牙利： $O(n^3/w)$ 。
 - 可持久化线段树优化建图 dinic： $O(n^2 \log n)$ 。
 - 线段树优化 hopcroft： $O(n^{1.5} \log n)$ 。
- 均可通过。
- 内存设 16MB，是为了防止一些经过优化的区间 dp 通过，正常实现的上述匹配算法，都没有空间的问题。

L - Aqre

- 如果我们能把盘面划分成若干个 1×4 的形状，那么每个 1×4 的矩形里面至少需要一个 0。
- 我们把整个盘面按照 $(i + j) \bmod 4$ 标号，每个 1×4 的矩形会占据 0123 各一个，所以 1×4 的矩形不会超过 0123 里面的最小值，并且当 $n, m \geq 4$ 的时候，一定能取到。
- 这样就得到了问题的上界，并且可以通过手玩发现，如果我们把整个盘面按照 4×4 循环构造，一定取到上界。
- 所以可以枚举 4×4 如何构造的，然后通过本地测试是否联通以及是否取到了最大值，输出一个满足条件的解即可。

L - Aqre

- 对于 $n \leq 3$ 的情况，上面的证明是有问题的，具体的是划分成 1×4 部分，因为不能竖着切，所以按 0123 标号的那个下界取不到。
- 又因为连通性，比如 3×7 我们只能划分出 6 个 1×4 的，如果要取到这个下界，会有两个独立的 3×3 ，是不合法的。
- 经过一些讨论和分析，只有 $3 \times (4k + 3)$ 需要特殊构造一下，否则按照前面的那个方法还是能构造出最优解。

M - Dining Professors

- 考虑每盘菜辣或者不辣，分别能有多少人吃。
- 我们希望能吃的人尽量多，所以就按照每盘菜如果是辣的，能吃的人数，从大往小贪心放即可。
- 时间复杂度 $O(n)$ 。