2025 ICPC 国际大学生程序设计竞赛
亚洲区域赛（南京站）

SUA 程序设计竞赛命题组

2025 年 11 月 9 日

概况

- 预期难度：
  - C $<$ K $<$ FGIM $<$ BEHJ $<$ AL $<$ D
- 实际通过人数排序：
  - TODO。

C. Distributing Candies

题意

- 给定 n，求正整数 a 和 b 使得 $a+b=n$，且两者均能整除 n。
- $n \leq 10^{18}$。

- 若 n 是奇数，只能分成奇数加偶数。奇数不能被偶数整除，因此无解。
- 若 n 是偶数，分成 $a=b=\frac{n}{2}$ 即可。

K. Xiangqi

题意

- 在一个象棋棋盘上面有一个马和一个车。马先手，问车是否能在有限步数内抓死马。

- 有向图博弈：对局面建图，从结束状态开始倒推：
  - 若一个点可以到达必败态，则它是必胜态；
  - 若一个点的所有出边都只能到达必胜态，则它是必败态；
  - 剩下不能确定的状态都不能在有限步数内结束。
- 实际上由于棋盘足够大，可能会想到：如果车不能一步抓死马，则永远不能抓死马。这个结论确实是对的。

F. Bitwise And Path

题意

- 给定一张初始没有边的 n 个节点的无向图，进行 q 次操作。
- 每次操作要么添加一条带权的边，要么求两点间的一条路径，使得路径上所有边权的 bitwise and 最大。
- $n \le 10^3$，$q \le 10^6$，边权 $V \le 2^{12}$。

- 如果有并查集 $\mathbb{S}(m)$ 记录二进制边权 m 以及它所有超集的连通关系，我们就能通过按位枚举答案的方式，处理每次询问。
- 怎么维护这种并查集呢？暴力即可。连接 (u, v) 之间权值为 m 的边时，通过递归的方式，每次去掉一个 bit 枚举 m 的子集并连接并查集。如果发现已经连通了，就不再继续递归。
- 每次成功的连接可能会带来 $\log V$ 个失败的检查，所以复杂度 $\mathcal{O}(\alpha n V \log V)$。再加上按位枚举答案的复杂度，总体复杂度 $\mathcal{O}(\alpha(nV \log V + q \log V))$。

G. Bucket Bonanza

题意

- 给定 n 个水桶，第 i 个水桶能盛 v_i 的水，每秒漏 l_i 的的水。
- 在第 0 秒开始前你可以任意地合并水桶，合并出的水桶容量取最大值，漏水取最小值。
- q 组询问，每次问第 0 秒加满水后第 t_i 秒最多能留多少水。
- $n \le n, q \le 2 \times 10^5$。

G. Bucket Bonanza

- 先不考虑有的水桶会漏完的情况。合并两个水桶等价于删除两者间较小的容量和较大的漏水率。可以发现答案和水桶容量-漏水对应关系无关，只和容量集合和漏水集合有关。
- 对于每次询问，考虑贪心，每次尝试合并容量最小的水桶和漏水最多的水桶，直到最小容量大于最多漏水量。如果容量最小和漏水最多是同一个水桶，这个水桶不可能储水，等价于删除。
- 接下来考虑有的水桶会漏完的情况，此时把漏完的水桶和其它桶合并，答案不会变得更差。所以至少存在一个最优答案是水漏不完的（或只保留 0 个水桶），直接用之前的贪心解决即可。
- 每次询问二分删多少个水桶即可。

I. Chi Fan

题意

- n 天，Grammy 有 m 元预算。每天可以选择：
- 花 b_i 自己独自吃饭，获得 a_i 的满意度；或者花 d_i 和队友出门吃饭，获得 c_i 的满意度。
- 两个队友 A 和 B 每天都会出门吃饭，出门吃饭需要打车，如果这天 Grammy 不来，A 和 B 各有概率支付这次车费。
- 如果 Grammy 选择出门吃饭，且他上次出门至今，两个队友都付过打车费，那么他要付这次车费，否则他可以让他的两个队友按上一条规则付车费。
- 问在保证预算够花的情况下，最优策略下 Grammy 能获得的最大满意度。
- $n \le 2 \times 10^3, m \le 5 \times 10^3$。

I. Chi Fan

- 因为要在保证后面的预算够用的条件下计算最大满意度期望，考虑对这 n 天倒着做 dp。
- 设 $f[p][q][a][b]$ 表示第 p 天开始有 q 的预算，之前 A 和 B 分别是否已经打过车了，到第 n 天结束，最大的满意度的期望。
- 分别枚举当天是否出去吃，若出去吃是否需要打车的几种情况做转移即可。
- 时间复杂度 $O(nm)$

H. Pen Pineapple Apple Pen

题意

- 给定一个字符串 S，从左往右依次选出 4 个不重叠的非空子串，使得第一个子串和第四个子串相等，并且第三个子串是第二个子串的严格后缀，不同的位置算不同的方案，求方案数对 $998\,244\,353$ 取模的结果。
- $|S| \le 5000$。

- 记 $f_{l,r}$ 为第一个子串在位置 l 结束、第四个子串从位置 r 开始的方案数，$g_{l,r}$ 为第二个和第三个子串在区间 $[l,r]$ 内的方案数，答案就是 $\sum_{1<l\le r<n} f_{l-1,r+1} \times g_{l,r}$。

H. Pen Pineapple Apple Pen

- 如果 S 的一个严格子串同时是 S 的前缀和后缀，称这个子串是 S 的 border。
- 对于 $f_{l,r}$，枚举 r，对 $S[r,n]$ 跑 KMP 求出 next 数组，其对应 $S[r,n]$ 每个前缀的最长 border，沿 next 数组跳到空串的次数就是 border 个数。
- 再对 $S[1,l]$ 利用刚才求出的 next 数组跑 KMP 匹配可以求出每个 $S[1,l]$ 的最长后缀等于某个 $S[r,n]$ 的前缀，$f_{l,r}$ 就是对应前缀的 border 个数。

H. Pen Pineapple Apple Pen

- 对于 $g_{l,r}$，先求出 $h_{l,r}$ 表示第三个子串在位置 r 结束、第二个子串中与第三个子串相同的后缀从位置 l 开始的方案数，那么 $g_{l,r} = \sum_{l \le i \le j \le r} (i-l) h_{i,j}$，二维前缀和优化即可。
- 实际上 $h_{l,r}$ 就是 $S[l,r]$ 长度不超过一半的 border 个数，一个结论是长度超过一半的 border 长度构成等差数列，对 $S[l,n]$ 跑 KMP 求出 next 数组，利用周期性可以快速从 $S[l,r]$ 跳到长度超过一半的最短 border，再沿 next 跳一次即可。
- 时间复杂度 $O(n^2)$。

B. What, More Kangaroos?

题意

- 给定 n 只数轴上位于正数坐标的袋鼠，有 4 种按钮同时控制它们在数轴上跳，每种按钮可以按任意多次，最大化操作后位于负数坐标的袋鼠个数。
- $n \leq 2 \times 10^5$。

- 简单来说就是给定 n 个不等式 $a_ix + b_iy + c_i < 0$ $(c_i > 0)$，找一组整数 (x, y) 使得最多的不等式成立。
- $a_i = 0$ 且 $b_i = 0$ 的不等式一定不成立，直接忽略，现在每个不等式对应一个半平面，要找一个属于最多半平面的整点。

- 对于一个点 $(x, y) \neq (0, 0)$，取一个充分大的正整数 t：
  - 如果 $a_i x + b_i y \geq 0$，总有 $a_i(tx) + b_i(ty) + c_i \geq c_i > 0$；
  - 如果 $a_i x + b_i y < 0$，由于 t 充分大，有 $a_i(tx) + b_i(ty) + c_i < 0$。
- 这说明选定方向后，选取充分远处的点不会比选取近处的点差，并且 c_i 对充分远点不会构成限制。
- 不妨令 $c_i = 0$，那么每个半平面覆盖了一个极角开区间，极角排序求出最大覆盖次数即可。
- 由于整点的极角是稠密的，总能在某个取到最大覆盖次数的极角开区间里找到一个整点的极角。
- 时间复杂度 $\mathcal{O}(n \log n)$。

M. Many Convex Polygons

题意

- 按照逆时针顺序给出凸 n 边形的顶点，选 k 个不同的顶点顺次连成凸 k 边形，对所有选法计算求凸 k 边形面积之和的两倍，对每个 $k = 3, 4, \ldots n$ 计算答案对 $998\,244\,353$ 取模后的结果。
- $3 \le n \le 2 \times 10^5$。

- 记选出的 k 个点为 $P(i_0), P(i_1), \cdots, P(i_{k-1})$，那么凸 k 边形面积的两倍是

$$\sum_{j=0}^{k-1} P(i_j) \times P(i_{(j+1) \bmod k}) = \sum_{j=0}^{k-1} (x_{i_j} y_{i_{(j+1) \bmod k}} - y_{i_j} x_{i_{(j+1) \bmod k}})$$

M. Many Convex Polygons

- 考虑线段 $P(i)P((i+d) \bmod n)$ 作为凸 k 边形的边，线段右侧的 $d-1$ 个点都不能作为凸 k 边形的顶点，还要从左侧的 $n-1-d$ 个点中选出 $k-2$ 个点作为凸 k 边形的顶点，那么这条线段对 ans_k 的贡献是

$$\binom{n-1-d}{k-2}\left(x_i y_{(i+d) \bmod n} - y_i x_{(i+d) \bmod n}\right)$$

- 对 i 求和，记 $f_d = \sum_{i=0}^{n-1}\left(x_i y_{(i+d) \bmod n} - y_i x_{(i+d) \bmod n}\right)$，使用 NTT 优化循环差卷积即可求出所有 $f_1, f_2, \ldots, f_{n-2}$。
- 于是有 $ans_k = \sum_{d=1}^{n-1}\binom{n-1-d}{k-2} f_d = \frac{1}{(k-2)!}\sum_{d=1}^{n-k+1}\frac{(n-1-d)!f_d}{(n-k+1-d)!}$，使用 NTT 优化差卷积即可求出所有 $ans_2, ans_3, \ldots, ans_n$。
- 时间复杂度 $\mathcal{O}(n \log n)$。

E. Cyan White Tree

题意

- 给定一棵 n 个点的有根树，其中 1 是根，每个点被染成青色或者白色。对树上的一条简单路径，记青点个数为 c、白点个数为 w，路径的价值是 $(c+w)-2|c-w|$，对每个 i 求出所有满足深度最小的点是 i 的简单路径中的最大价值。
- $n \le 4 \times 10^5$。

- 先处理 $c \ge w$ 的情况，此时路径的价值是 $3w-c$，将颜色取反再做一次就能处理 $c \le w$ 的情况。

E. Cyan White Tree

- 记子树 u 的高度为 h_u，考虑 $f_u[i]$ 表示从 u 往下的满足 $c-w \ge i$ 的所有简单路径中的最大价值，那么当 $i > h_u$ 时总有 $f_{u,i} = -\infty$，当 $i < -h_u$ 时总有 $f_u[i] = f_u[-h_u]$，因此只需要维护 $f_u[-h_u, h_u]$。
- 考虑自底向上转移，对于非叶子节点 u，任取一个子树高度最大的儿子节点 son_u 把 f_{son_u} 继承过来，加上 u 的贡献之后作为 f_u，然后用 $f_u[0]$ 更新 ans_u。
- 在给 son_u 往下的路径加上 u 的贡献时，需要先对 f_{son_u} 数组整体做偏移、对整体加一个值，可以使用 std::deque 结合全局加减标记实现。还要加入路径只有 u 的情况，这需要从更新的位置（$f_u[-1]$ 或者 $f_u[1]$）开始往前扫一遍后缀 max，但是完整扫一遍不能接受，这里先扫到 $f_u[0]$ 以确保 $f_u[0, h_u]$ 的值是对的。

E. Cyan White Tree

- 现在考虑 u 的其他子树 v，由于之前的过程里 f_v 可能没有完整更新，可以先 $\mathcal{O}(h_v)$ 扫一遍 f_v 的后缀 max 以确保 $f_v[-h_v, h_v]$ 的值是对的，顺便也扫一遍 $f_u[-h_v, h_v]$ 以确保 $f_u[-h_v, h_u]$ 的值是对的。
- 考虑枚举 i 用 $f_u[-i] + f_v[i]$ 更新 ans_u，显然 $i > h_v$ 没有意义，如果 $i < h_v$ 那么 $f_v[i] = f[h_v]$ 但是 $f_u[-i] \leq f_u[-h_v]$，因此只需要枚举 $i \in [-h_v, h_v]$。
- 再把 f_v 转移给 f_u，类似地需要先给 v 往下的路径加上 u 的贡献，然后用加上贡献后的 $f_v[i]$ 更新 $f_u[i]$，最后再扫一遍 $f_u[-h_v, h_v]$ 以确保 $f_u[-h_v, h_u]$ 的值是对的。
- 沿用长链剖分的分析，时间复杂度 $\mathcal{O}(n)$。

J. Trajan Algorithm

题意

- 把求割点的 Tarjan 算法里，计算 $low[u]$ 的方式全部改成 $low[u] = min(low[u], low[v])$，会导致哪些点被判错？

- 假算法只会让 low 变得更小，因此只会把割点判漏，而不是反过来。
- 被修改的 case 是返祖边 $v \to u$，其中 u 是被判漏的割点，因此这个错误的 low 是从 $low[u]$ 传递过来的。因此，当 $low[u] < dfn[u]$ 时，u 会被判漏。
- 什么情况下 $low[u] < dfn[u]$？那就是 u 也有返祖边。

J. Trajan Algorithm

- 总结：割点 u 被判漏，若有返祖边指向它，以及有从它离开的返祖边。即 u 至少属于两个点双连通分量，这两个分量各自至少有两条边连接 u。
- 但是，如果 u 还属于另一个点双连通分量，这个分量只有一条边连接 u，那么 u 会因为这个连通分量成为割点。所以还要排除这个情况。
- 复杂度 $\mathcal{O}(n+m)$。
- Fun Fact 1：本题原本是给 2020 年南京站准备的。
- Fun Fact 2：命题人初学 Tarjan 后写了三年假算法，没被卡过。

L. Regional Champion

题意

- 在平面上画 n 个圆，m 个三角形，k 条直线，使得平面被划分成尽可能多的区域。
- 需要输出最大划分区域数以及构造方案。

L. Regional Champion

- 连通图中有 $F = E - V + 2$。而每产生一个新的交点会新增两条边，所以交点数越多越好。
- 圆视为一个点绕过圆环连向自身，$V = 1, E = 1$。
- 三角形有三个端点和三条边，$V = 3, E = 3$。
- 直线视为一个共享的无穷远点连向自身，$V = 1, E = 1$。
- 这部分的 $E - V$ 为 $k - [k \neq 0]$。

L. Regional Champion

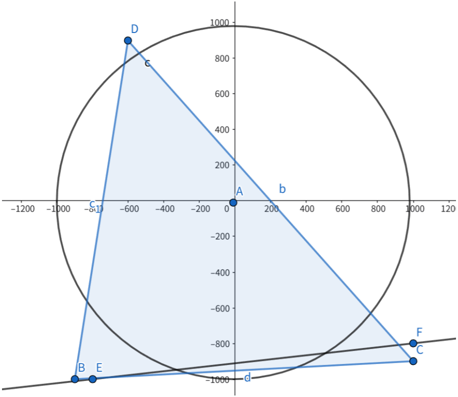
- 这些图形彼此之间能产生的最大交点数如下。

	圆	三角形	直线
圆	V+=2	V+=6	V+=2
三角形		V+=6	V+=2
直线			V+=1

- 这部分的 $E-V$ 为 $2\binom{n}{2}+6\binom{m}{2}+\binom{k}{2}+6nm+2mk+2nk$。
- 所以总的区域数为
  $F=2\binom{n}{2}+6\binom{m}{2}+\binom{k}{2}+6nm+2mk+2nk+k-[k\neq 0]+2$。

L. Regional Champion

- 构造时需要让这些图形两两之间达到最大交点数，并且所有的这些交点互不重合（最好和三角形的端点也不重合）。
- 先构造一个图形比较大的 $n = m = k = 1$ 的版本。



L. Regional Champion

- 然后尝试扰动这三个基本图形，产生更多的圆、三角形、直线。
- 比如构造圆形为 $(x-i)^2+(y-j)^2=990^2$，其中 $-10 \le i,j \le 10$。
- 三角形为 $(B+(i,0), C+(0,i), D+(-i,-i))$，其中 $1 \le i \le 100$。
- 直线为 $(E+(i,0), F+(0,i))$，其中 $1 \le i \le 100$。
- 然后取出这个 list 里面的前 n,m,k 个。

A. Wow, It's Yesterday Six Times More

题意

- 给定一个 n 行 m 列的网格，恰好有一个格子是洞，但并不知道是哪个格子，其他格子都是空地，一开始每个空地都有一只袋鼠。
- 你需要找出是洞的格子，允许交互最多 $(n+m+10)$ 次，每次可以让所有袋鼠同时往上/下/左/右移动一步，走出网格或走到洞里的袋鼠就被移除，交互器返回剩下袋鼠的数量。
- $3 \leq n, m \leq 30$。

- 实际上能做到 $\min(r, n+1-r) + \min(c, m+1-c) + \mathcal{O}(1)$ 次交互，假设洞位于第 r 行第 c 列的格子。

A. Wow, It's Yesterday Six Times More

- 根据移动后的返回剩下袋鼠的数量可以知道这次移动的袋鼠减少量，以此进行决策。以下给出标程使用的决策方案：
- 首先连续下移最多 $\lfloor \frac{n}{2} \rfloor$ 次直到第 k 次下移的减少量 $\leq m$，那么洞在第 k 行或者第 $m-k$ 行。当且仅当 n 是奇数时可能前 $\lfloor \frac{n}{2} \rfloor$ 次操作的减少量都 $> m$，此时 $k = \frac{n+1}{2}$。
- 先讨论 n 是奇数且 $k = \frac{n+1}{2}$ 的情况，那么洞一定在第 k 行。此时已经下移了 $\lfloor \frac{n}{2} \rfloor = k-1$ 次，除了洞所在的列，每一列最下边 $n-k+1$ 行都有袋鼠。此时开始连续左移直到第 t 次左移的减少量 $\leq n-k+1$，如果减少量 $\leq n-2k+2$ 那么洞在第 t 列，否则在第 $m+1-t$ 列。

A. Wow, It's Yesterday Six Times More

- 对于剩下的情况，从初始状态开始做了 k 次下移，此时洞要么在第 k 行，要么在第 $n-k+1$ 行，先讨论 $k>1$ 的情况。
- 首先左移一次：
  - 如果减少量是 $n-2k$，那么洞在 $(k,1)$；
  - 如果减少量是 $n-2k+1$，那么洞在 $(n+1-k,1)$；
  - 如果减少量是 $n-k+1$，那么洞在第 $n+1-k$ 行，且不在第 1 列和第 m 列。此时开始连续左移直到第 t 次左移的减少量 $\le n-k$，如果减少量 $\le n-2k+1$ 那么洞在第 $t+1$ 列，否则在第 $m-t$ 列。
- 如果不属于上述三种情况，上移一次，如果减少量 >0，那么洞在第 k 行，且不在第 1 列和第 m 列，此时也转入与上述完全相同的连续左移的过程。
- 否则洞肯定在第 m 列，先右移两次，再下移一次，如果减少量 $=0$ 那么在第 k 行，否则在第 $n+1-k$ 行。

A. Wow, It's Yesterday Six Times More

- 现在讨论 $k = 1$，也就是从初始状态开始只做了一次下移，此时洞要么在第 1 行，要么在第 n 行。
- 首先左移一次：
  - 如果减少量是 $n-2$，那么洞在 $(1,1)$；
  - 如果减少量是 n，那么洞在第 n 行，且不在第 1 列和第 m 列。此时开始连续左移直到第 t 次左移的减少量 $< n$，然后再下移一次，如果减少量 $> m-2t-2$ 那么洞在第 $t+1$ 列，否则在第 $m-t$ 列。
- 如果不属于上述两种情况，上移一次，如果减少量 > 0，那么洞在第 1 行，且不在第 1 列和第 m 列。此时开始连续左移直到第 t 次左移的减少量 $< n$，然后再上移一次，如果减少量 $> m-2t-2$ 那么洞在第 $t+1$ 列，否则在第 $m-t$ 列。
- 否则再上移一次，如果减少量 $= m-2$，那么洞在 $(1,m)$，否则再左移一次，如果减少量 $= n-2$，那么洞在 (n,m)，否则洞在 $(n,1)$。
- 至此所有情况都讨论完了。

题意

- 有 k 个大小为 n_i 的树，每个树的某个节点上有个小球。
- 每次可以花 1 的代价，选一个树，该树的小球随机游走一步。
- 只要存在一个小球游走到根节点，游戏胜利。
- 求最优决策下，获胜的最小代价的期望。
- $k, n_i \leq 300$。

D. Fallleaves01 And Golf

- 我们先解决一个重要的问题：击球策略是什么？
- 先从另一个模型入手。考虑在一棵树上随机游走，一开始你站在节点 s，目标是走到节点 1。每次，你有两种策略可以选择：
  - 你可以花一块钱，等概率走到某一个相邻的节点。
  - 直接退出游戏。
- 若你走到节点 1，游戏会立刻终止，且你会获得 r 元钱的奖励。
- 我们发现，若 r 小于某个阈值，你一开始（在 s 时）就会停止游戏，若 r 大于该阈值，你会选择游走。当 r 等于该阈值时，选择两种策略的期望收益是相等的。

D. Fallleaves01 And Golf

- 称这个阈值为 grade_s。由于直接退出游戏的期望收益是 0，因此当 $r = \text{grade}_s$ 时继续游戏的期望收益也为 0。由于退出的期望收益是 0，从 s 点开始这个游戏的期望收益也是 0，退出可以看成是瞬移回 s 点重开游戏。那这个游戏也有另一种理解：我们在 s 点进行随机游走，可以在任意时刻瞬移回 s。此时随机游走到终点的期望时间为 grade_s。利用这种理解，可以在 $O(n^2)$ 的时间内求出一棵树内每个节点的 grade。
- 回到原先的题目，策略是每次选择 grade 最小的一个球进行击打。
- 考虑此时的期望击球数如何计算。我们假设每颗球的游走路径都已经确定。设最终游走到终点的球经过的节点的最大的 grade 为 G。那么，此时其他的球一定是停止在其路径的第一个 grade 大于 G 的节点上。我们枚举取到 G 的节点，对其他节点求出相应的条件期望即可。
- 时间复杂度 $O(n^2k)$。

Thank you!