

欧洲信息学竞赛题目选讲

NOI2023 冬令营

钟诚

目录

- 课程简介
- 欧洲信息学竞赛概述
- 欧洲信息学竞赛的命题风格
- 题目选讲
- 总结

课程简介

- 中国自1984年举办NOI、1989年参加首届IOI以来，在信息学竞赛领域取得了十分优异的成绩。NOI系列活动的OI科技树也枝繁叶茂、蓬勃发展，不断有新的数据结构、算法或解题模式涌现。
- 近年来，IOI和其它各级各类国际比赛也进入了高速发展期，产生了许多形式新颖的题目，其中部分类型和考点的题目在NOI系列活动中出现的比重较小。
- 本次课程希望能够抛砖引玉，通过对欧洲各大信息学竞赛的介绍和题目选讲，帮助同学们开拓思维、取长补短，了解更多国际比赛的题目风格，以及相应的解题策略，鼓励同学们今后为我们的OI科技树添砖加瓦。
- 讲者：钟诚，曾任NOI科学委员会学生委员，现为Google苏黎世软件工程师，2022年参与了欧洲女生信息学奥林匹克（EGOI）的命题工作。

欧洲信息学竞赛概述

- 各国的 “NOI”
- 中欧信息学奥林匹克 CEOI
 - 创办于1994年，是欧洲历史最悠久的历史学竞赛
- 欧洲女生信息学奥林匹克 EGOI
 - 有45个国家（近一半参与IOI的国家）参与2022年的比赛，是欧洲参与最广泛的信息学竞赛
- 欧洲初中信息学奥林匹克 EJOI
- 巴尔干半岛信息学奥林匹克 BOI、巴尔干半岛初中信息学奥林匹克 JBOI
- 波罗的海信息学奥林匹克 BOI
- 克罗地亚信息学公开赛 COCI
- 比荷卢联盟信息学奥林匹克 BxOI

欧洲信息学国家队选拔训练周期——以瑞士为例

- 上一年9月~11月：SOI第一轮，全部为作业形式
 - 分为普及组、提高组，各有5~6个IOI形式的题目，两个组别都有晋级下一轮的名额
- 上一年10月左右：在主要城市开展多语种的普及性讲座
- 2月：冬令营，纯培训（Social），不设选拔性考试
- 3月：SOI第二轮
 - 作业：从第一轮结束后开始，学习12个专项并完成习题（DFS、BFS、拓扑排序、二分法、前缀和、DP、Subsetsum、Dijkstra、最小生成树、并查集、凸包、线段树）
 - 第一试：形式为上机，与IOI类似
 - 第二试：形式为笔试，主要为算法设计和复杂度分析
- 5月：国家队选拔赛
- 6月~9月：参加各项国际比赛（EGOI、CEOI、IOI）

欧洲信息学竞赛的命题风格

- 对高级数据结构和算法的要求较低
 - 欧洲竞赛更注重思维方面的考察；除法国等个别国家外，几乎不超过十年前NOIP提高组的考察范围，可参考下一页所附的瑞士信息学竞赛的主要考察范围
 - 即使是在国家级的比赛中，也有不少不超过CSP-J范围的签到题，以此鼓励更多的同学参与
- 交互题占的比重较高
 - 和NOI系列活动的题目相比，欧洲竞赛交互题 / 构造性题目的比重较高，而传统数据结构和算法题的比重较低
- 有包括笔试在内的更多考察形式
 - 由于参与的人数较少（或极少），使得人工阅卷成为可能
- 对“卡常”的容忍度较低

附：瑞士信息学竞赛主要考察范围

- 排序、二分法
- 数据结构：栈、队列、列表、set、map、优先队列、双端队列、二叉堆、区间树、线段树、并查集、有理数、高精度、四叉树、树状数组
- 图算法：DFS、BFS、连通分量、拓扑排序、最短路 (Dijkstra, Bellman-Ford, Floyd)、最小生成树 (Kruskal, Prim, 割点, Hierholzer)
- 动态规划：前缀和、Levenshtein距离、最长公共子序列、最长上升子序列、矩阵链乘积、Maximum Empty Rectangle
- 字符串：KMP
- 计算几何：略
- 组合数学：排列、子集、生成下一个字典序的排列
- 数论：最大公约数、最小公倍数、线性同余方程、质因数分解、埃氏筛素数法
- 其它：容斥原理、最近公共祖先、稳定婚姻问题、RMQ问题

题目选讲

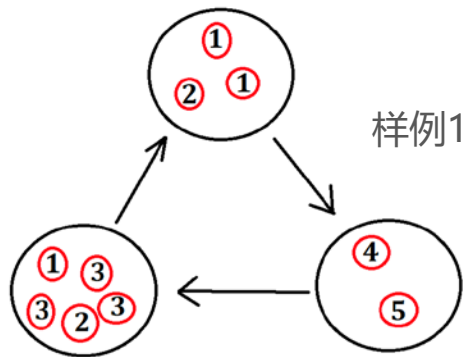
部分题目可能有多种解法，欢迎同学们踊跃发言！

三角形演讲（瑞典程序设计奥林匹克 PO-Final 2022）

一个班级有N个同学，分为3组，第1组的给第2组的做演讲，第2组的给第3组的做演讲，第3组的给第1组的做演讲。每个同学都有给很多人演讲的雄心壮志——编号为i的同学希望其听众至少有A[i]个同学。

输入N和数组A[1..N]，请将所有的同学分为3个组，以满足所有同学的要求；或输出NO表示不存在符合要求的分组方式。

样例输入1	样例输出1	样例输入2	样例输出2
10 1 3 1 3 3 2 4 1 5 2	YES 3313332121	3 1 2 2	NO



三角形演讲（瑞典程序设计奥林匹克 PO-Final 2022）

数据范围： $3 \leq N \leq 5 * 10^5$, $1 \leq A[i] \leq N$

Subtask 1 (14分) : $A[1] = A[2] = \dots = A[N]$

Subtask 2 (16分) : $N \leq 10$

Subtask 3 (11分) : $A[i] \leq 3$

Subtask 4 (23分) : $N \leq 3000$

Subtask 5 (36分) : 无其它限制

玩具设计（欧洲女生信息学奥林匹克 EGOI 2022）

你在一个设计玩具的公司工作。某玩具有 n 个引脚，在其内部，某些引脚之间有电线相连。这些电线对外不可见，只有一个测试器能检测引脚 i 和引脚 j 是否直接或间接相连。我们把这样一套连接方案称为玩具的一个“设计”。初始时记为0号设计。

调用测试器的函数如下：

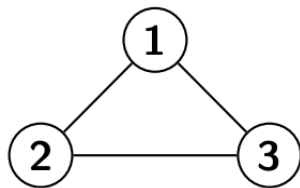
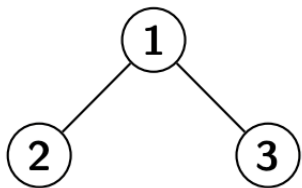
```
int Connected(int a, int i, int j);
```

表示询问第 a 号设计中，引脚 i 和引脚 j 是否直接或间接相连。该测试器返回一个整数，如果相连，则直接返回 a 。如果不相连，则将创造一个新的设计，该设计是第 a 号设计的基础上加上 i 与 j 连的边，设计的编号为下一个可用的编号，并返回这个编号。无论如何，不会修改原先的第 a 号设计。

玩具设计（欧洲女生信息学奥林匹克 EGOI 2022）

你的任务是还原第0号设计。

诚然，有时候我们可能无法确定所有的连接情况。如果两个设计无法通过测试器区分，即对于任意的 i, j ，测试器都会给出相同的回答，那么就认为它们是相同的设计。例如下面两个设计就被认为是相同的：



请返回任何一个与第0号设计相同的设计。

玩具设计（欧洲女生信息学奥林匹克 EGOI 2022）

样例交互：

选手操作	评测机操作	说明
	ToyDesign(4, 20)	$n = 2$, $\text{max_ops} = 20$
Connected(0, 1, 2)	Returns 1.	在第0号设计中，引脚1和引脚2不直接或间接相连，因此创造新的第1号设计。
Connected(1, 3, 2)	Returns 2.	在第1号设计中，引脚3和引脚2不直接或间接相连，因此创造新的第2号设计。
Connected(0, 3, 4)	Returns 0.	在第0号设计中，引脚3和引脚4直接或间接相连。
DescribeDesign({{3, 4}})	-	返回一个与第0号设计相同的设计：仅有引脚3和引脚4之间直接相连。

玩具设计（欧洲女生信息学奥林匹克 EGOI 2022）

数据范围： $2 \leq n \leq 200$

Subtask 1（10分）： $n \leq 200$, $\text{max_ops} = 20\,000$

Subtask 2（20分）： $n \leq 8$, $\text{max_ops} = 20$

Subtask 3（35分）： $n \leq 200$, $\text{max_ops} = 2\,000$

Subtask 4（35分）： $n \leq 200$, $\text{max_ops} = 1\,350$

其中 max_ops 是调用测试器的最多次数

信息传递（波罗的海信息学奥林匹克 BOI 2022）

你在一艘潜水艇里，想给你的助理发送 $1 \sim N$ 中的一个数。每次你只能发送1比特的信息（即0或1），但由于各种因素的影响，实际收到的信息可能有误。你会在每次发送完成后，得知实际被收到的信息。该系统保证任何连续两次发送中，至少有一次是正确的。

你的助理需要根据收到的信息，找出两个数，确保你想发送的 $1 \sim N$ 中的数是其中之一。

交互方式：你需要实现两个函数

- `void encode(int N, int X)`
- `std::pair<int, int> decode(int N)`

信息传递（波罗的海信息学奥林匹克 BOI 2022）

其中，`encode`函数扮演“你”的角色，有两个参数`N`和`X`，希望发送 $1 \sim N$ 中的一个数`X`。每次发送1比特的信息时，你需要调用交互库中的函数：`int send(int s)`，其中`s`为你发送的信息（0或1），函数的返回值是你的助理实际收到的信息。

而`decode`函数扮演“你的助理”的角色，有一个参数`N`，负责接收和解密信息。每次接受1比特的信息时，需要调用交互库中的函数：`int receive()`，函数的返回值是实际收到的信息。最后，`decode`函数的返回两个整数 $\langle a, b \rangle$ ，表示解密出“你”想发送的数`X`一定是`a`和`b`中的一个。注意，交互库并不会告诉你一共有多少信息需要接收，你的程序需要自行判断何时停止并`return`答案。

信息传递（波罗的海信息学奥林匹克 BOI 2022）

样例交互：

encode(1337, 42)		
操作	返回值	说明
调用send(1)	0	实际收到的信息错误
调用send(0)	0	实际收到的信息正确 (根据题目约定, 本次必然正确)
调用send(1)	1	实际收到的信息正确
调用send(1)	0	实际收到的信息错误

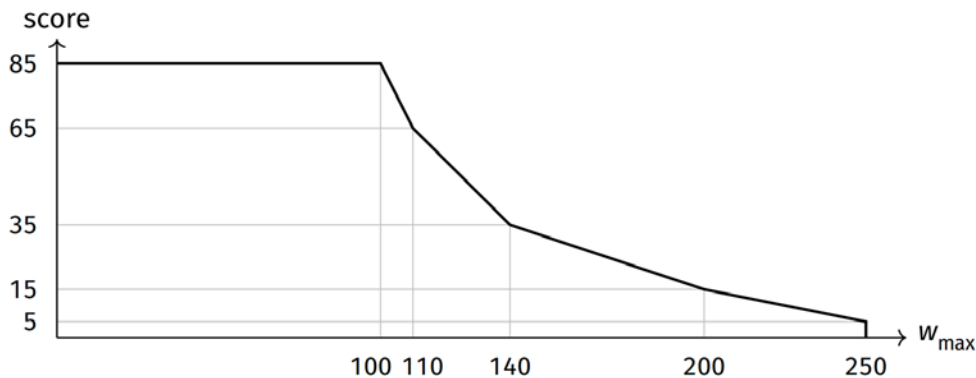
decode(1337)		
操作	返回值	说明
调用receive()	0	实际收到的信息为0, 尽管你发送的是1
调用receive()	0	实际收到的信息为0
调用receive()	1	实际收到的信息为1
return {1337, 42}	0	解密成功, 程序结束

信息传递 (波罗的海信息学奥林匹克 BOI 2022)

数据范围: $3 \leq N \leq 10^9$

Subtask 1 (15分) : $N = 3$

Subtask 2 (85分) : 无其它限制。在多个测试数据中, 你发送信息最多的是 w_{\max} 次, 则你的得分如下图所示。特别地, 如果对于任何数据都发送不超过100次信息, 可以得到满分。



信息传递（波罗的海信息学奥林匹克 BOI 2022）

提示（这不是正式题面的一部分，但原题面理解起来比较困难，因此添加了本提示）：

首先，样例中的步骤仅展示了交互模式，而没有算法上的实际意义。事实上，当 $N = 1337$ 时，不可能仅发送4比特的信息即能确定 X 的值。另外，`decode`函数最后返回的数对 $\langle 1337, 42 \rangle$ 并不是唯一的正确返回值， $\langle 100, 42 \rangle$ ， $\langle 42, 200 \rangle$ 等任何包含42的数对均为正确答案。

之所以没有给一个算法上更有实际意义的样例，估计是想把这15分作为Subtask 1，留给选手。但其实这样是挺难理解题意的，所以在这里我们提供一个骗分解法，便于大家理解题意。

当 $N = 3$ 时，下述解法有很大概率能骗分成功。在`encode`函数中：

- 如果 $X = 1$ ，则调用100次`send(0)`
- 如果 $X = 2$ ，则调用100次`send(1)`
- 如果 $X = 3$ ，则调用100次`send(rand(2))`

信息传递（波罗的海信息学奥林匹克 BOI 2022）

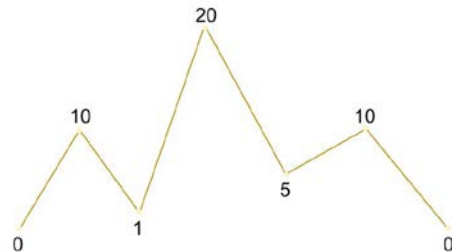
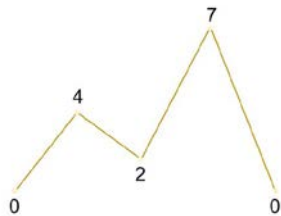
而在decode函数中，调用100次receive()：

- 如果有连续接收到两个1，由于任何连续两次发送中，至少有一次是正确的，则表明X不可能是1，返回<2, 3>
- 否则，如果有连续接收到两个0，同理X不可能是2，返回<1, 3>
- 否则，返回<1, 2>

如果测试数据比较水，且评测机每次都是随机决定收到的信息是否有误（当然，如果前一次收到的信息错误，那么下一次必然安排收到的信息正确），那么该算法有很大的概率将会通过。然而，如果decode函数收到的100次信息为1, 0, 1, 0, 1, 0, ...的话，理论上X可能等于1, 2, 3中的任何一个值，因此该算法并不完全正确。另外，请注意该算法并没有用到send函数的返回值。

登山者 (罗马尼亚信息学大师赛 RMI 2018)

Alice和Bob要爬的山可以用一系列折线表示 (左图为样例1, 右图为样例2), 最左端和最右端的高度均为0, 中间端点的高度均为正整数:



图中的数字代表高度。Alice从山的左端, Bob从山的右端开始爬山, 且保证在任何时刻, 必须处于相同的高度。初始时他们的高度 $h_1 = 0$, 每次改变爬山方向 (向上/向下) 时的高度分别为 h_2, h_3, \dots, h_p , 则爬山的体力值为 $|h_2 - h_1| + |h_3 - h_2| + \dots + |h_p - h_{p-1}|$ 。求Alice和Bob为了相遇所需的最小体力值。

登山者 (罗马尼亚信息学大师赛 RMI 2018)

样例输入1	样例输出1	样例输入2	样例输出2
5 0 4 2 7 0	11	7 0 10 1 20 5 10 0	48

样例1解释：Alice和Bob相向而行，爬到高度为4处，然后Alice和Bob都向右而行，爬到高度为2处，最后他们相向而行，爬到高度为7处相遇。

数据范围：

Subtask 1 (25分)：所有中间端点的高度互不相同

Subtask 2 (25分)： $N * H \leq 40\,000$ ，其中H表示最高的高度

Subtask 3 (50分)： $3 \leq N \leq 5\,000$ ，所有中间端点的高度介于1和 10^6 之间

寻找树根 (欧洲初中信息学奥林匹克 EJOI 2022)

这是一个交互题。给定一棵树中的所有节点和边，保证其中至少有一个节点与其它至少3个节点相连。你可以向交互库提出若干次下述模式的问题，并以此为依据最终找到树的根：

- 对于节点集合 $a_1 a_2 \dots a_m$ ，检查它们的最近公共祖先 (LCA) 是否在该集合内

交互方式：首先，输入 n ，并输入 $n-1$ 条树上的边。然后，每次询问的格式都是输出 “? $m a_1 a_2 \dots a_m$ ”，随后输入交互库给出的答案，YES代表它们的最近公共祖先就在集合内，NO表示不在。你的程序最多提出1000个问题，你的得分取决于问题个数的多少。

最后，当你找到树根时，输出 “! v ”，表示 v 就是要找的树根，并结束程序。

寻找树根 (欧洲初中信息学奥林匹克 EJOI 2022)

输入	输出
7 4 1 1 2 4 3 3 5 3 6 4 7	
	? 2 5 6
NO	
	? 3 6 3 5
YES	
	? 2 1 7
NO	
	? 2 4 6
YES	
	! 4

样例解释：本样例中，待寻找的根为节点4。

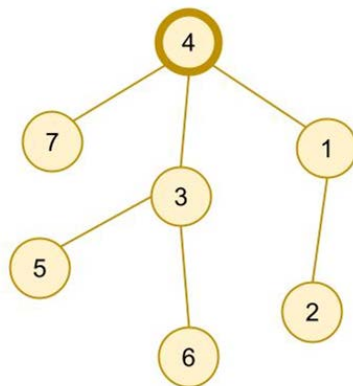
在第一次询问中，节点5、6的最近公共祖先为节点3，并不是5、6中的一个，因此交互库回答NO。

在第二次询问中，节点6、3、5的最近公共祖先为节点3，因此交互库回答YES。

在第三次询问中，节点1、7的最近公共祖先为节点4，因此交互库回答NO。

在第四次询问中，节点4、6的最近公共祖先为节点4，因此交互库回答YES。

最后，输出猜测的结果：树根为4，并结束程序。



寻找树根（欧洲初中信息学奥林匹克 EJOI 2022）

数据范围：

Subtask 1（7分）： $n \leq 9$

Subtask 2（10分）： $n \leq 30$

Subtask 3（83分）： $n \leq 500$

在前两个Subtask中，只要你的询问次数不超过1000，即可获得满分。在第三个Subtask中，记 k 为你的询问次数。如果 $k \leq 9$ ，则你将获得83分（满分），否则你将获得 $\lfloor \max(10, 83 \cdot (1 - (\ln(k - 6))/7)) \rfloor$ 分。

奖品（中欧信息学奥林匹克 CEOI 2022）

给定两棵树，称为1号树和2号树，它们各自都有编号为 $1, 2, \dots, N$ 的 N 个节点。每棵树上的每条边都有一定的正整数权值，但此时你并不掌握。

给定一个整数 K ，请你选择 $\{1, 2, \dots, N\}$ 的一个大小为 K 的子集，并询问 Q 次问题 (a, b) ，其中 a 和 b 都是你选定的子集中的元素，交互库将回答 $(d_1(l_1, a), d_1(l_1, b), d_2(l_2, a), d_2(l_2, b))$ ，其中 $d_t(x, y)$ 表示在 t 号树上节点 x 和节点 y 的距离， l_t 表示 t 号树上 a 和 b 的最近公共祖先。

在你提问完毕后，交互库会提出 T 个问题 (p, q) ，其中 p 和 q 都是你选定的子集中的元素，你需要回答 $(d_1(p, q), d_2(p, q))$ 。

奖品（中欧信息学奥林匹克 CEOI 2022）

交互方式：首先，第一行输入 N, K, Q, T ，第二行、第三行有 N 个数，依次代表1号树、2号树中编号为 $1, 2, \dots, N$ 的节点的父节点。

随后，你应当输出一行 K 个正整数，代表你选择的子集。然后，输出至多 Q 行，每行都是“? a b”，表示一个问题。此后输出一行“!”，表示提问完毕。

接着，输入若干行，每行都是四个整数 $d_1(l_1, a), d_1(l_1, b), d_2(l_2, a), d_2(l_2, b)$ ，依次为你的每个问题的答案。

然后再输入 T 行，每行两个整数 p, q ，表示交互库提出的一个问题。

最后，输出 T 行，每行两个整数 $d_1(p, q), d_2(p, q)$ ，依次为交互库提出的每个问题的答案。

奖品 (中欧信息学奥林匹克 CEOI 2022)

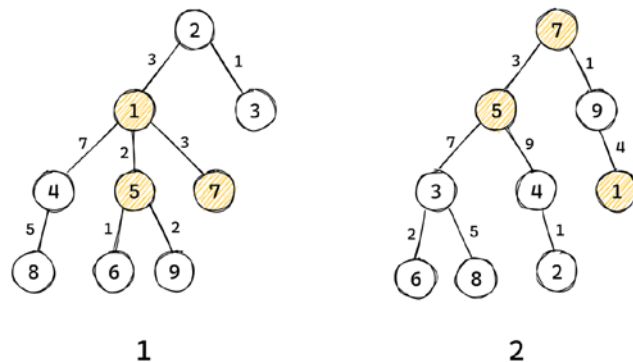
输入	输出
9 3 2 3 2 -1 2 1 1 5 1 4 5 9 4 5 5 7 3 -1 3 7	
	1 5 7 ? 1 5 ? 1 7 !
0 2 5 3 0 3 5 0 1 7 7 5 5 1	
	3 5 5 3 2 8

在此样例中，选择了子集 $\{1, 5, 7\}$ ，并询问了 $(1, 5)$ 和 $(1, 7)$ 这两个问题。

- 对于问题 $(1, 5)$ ，它们在1、2号树上的最近公共祖先分别是 $l_1 = 1$ ， $l_2 = 7$ ，因此交互库回答 $d_1(1, 1) = 0$ ， $d_1(1, 5) = 2$ ， $d_2(7, 1) = 5$ ， $d_2(7, 5) = 3$ 。
- 对于问题 $(1, 7)$ ，它们在1、2号树上的最近公共祖先分别是 $l_1 = 1$ ， $l_2 = 7$ ，因此交互库回答 $d_1(1, 1) = 0$ ， $d_1(1, 7) = 3$ ， $d_2(7, 1) = 5$ ， $d_2(7, 7) = 0$ 。

然后，交互库提出了三个问题 $(1, 7)$ ， $(7, 5)$ ， $(5, 1)$ ，其回答分别是：

- $d_1(1, 7) = 3$ ， $d_2(1, 7) = 5$
- $d_1(7, 5) = 5$ ， $d_2(7, 5) = 3$
- $d_1(5, 1) = 2$ ， $d_2(5, 1) = 8$



奖品（中欧信息学奥林匹克 CEOI 2022）

数据范围： $2 \leq K \leq 100\,000$, $1 \leq T \leq \min(K^2, 100\,000)$

Subtask 1（10分）： $N = 500\,000$, $Q = K - 1$, 两棵树一模一样（包括所有的边权）

Subtask 2（25分）： $N = 500\,000$, $Q = 2K - 2$

Subtask 3（19分）： $N = 500\,000$, $K = 200$, $Q = K - 1$

Subtask 4（22分）： $N = 1\,000\,000$, $K = 1\,000$, $Q = K - 1$

Subtask 5（24分）： $N = 1\,000\,000$, $Q = K - 1$

最大公约数（西班牙信息学奥林匹克 OIE 2022）

这是一个交互题，你要猜的是正整数 x 和 y 。你每次可以选择两个整数 a, b ，并询问交互库 $\text{mcd}(|x - a|, |y - b|)$ 的值，其中 mcd 是西班牙语里最大公约数的缩写。约定 $\text{mcd}(0, 0) = 0$ 。

交互方式：每次用`? a b`的格式输出一个问题，然后输入交互库的回答。如果交互库回答0，则表示已经猜中 x 和 y （因为这意味着 $x = a$ 且 $y = b$ ），你的程序应当结束。

输入		1		4		2		0
输出	? 0 0		? -1 0		? 5 2		? 3 4	

本样例中，要猜的 $x = 3$ ， $y = 4$ 。

最大公约数（西班牙信息学奥林匹克 OIE 2022）

数据范围： $1 \leq x, y \leq 10^{18}$ ，你选择的 a, b 需要满足： $-2 \cdot 10^{18} \leq a, b \leq 2 \cdot 10^{18}$ 。你最多可以询问250次，最后一次 $x = a$ 且 $y = b$ 的询问也计入其中。

Subtask 1 (5分) : $x, y \leq 15$

Subtask 2 (10分) : $x, y \leq 250$

Subtask 3 (15分) : x 和 y 的所有质因数都不超过5（即只可能是2、3、5）

Subtask 4 (30分) : $x, y \leq 10^9$

Subtask 5 (40分) : 没有其它限制条件

对于每个Subtask，如果你询问不超过125次，则可获得满分，否则根据询问次数 q ，得分要乘以一个系数：

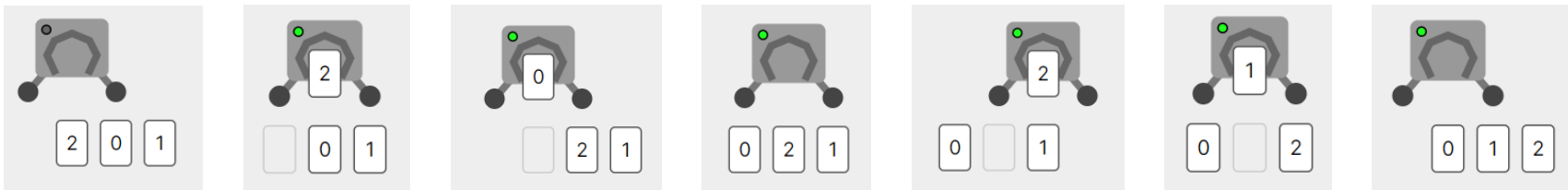
$$M(q) = \begin{array}{ll} 0 & q > 250 \\ 0.7 & 250 \geq q \geq 225 \\ 0.7 + (225 - q) / 500 & 225 > q > 125 \\ 1.0 & 125 \geq q \end{array}$$

爪式排序（瑞士信息学奥林匹克 SOI 2021/2022）

Stofl老鼠设计了一套爪式排序的机器。这个机器有 $0, 1, \dots, N - 1$ 这些位置的槽，以及一个爪子。初始时每个槽里有一张卡片，爪子在位置0上方且没有卡片。

机器有两种操作：右移、左移。在右移/左移时，将爪子里当前握住的卡片（可能为空）与当前位置的槽里的卡片（也可能为空）交换，并把爪子（带着它握住的卡片）向右/左移动一个位置。请通过这些操作，对 N 张卡片从小到大排序。

例如，当 $N=3$ ，将2 0 1进行排序，一个可行的操作顺序为：右-左-右-右-左-左。



爪式排序（瑞士信息学奥林匹克 SOI 2021/2022）

数据范围： $1 \leq N \leq 300$ ， K 代表最多进行几次操作

Subtask 1（15分）：输入操作顺序，模拟操作结果

Subtask 2（10分）： $N = 3$

Subtask 3（20分）： $N \leq 101$ ， $K \leq 10\,000\,000$

且已知原序列为逆序排列，即 $N - 1, N - 2, \dots, 1$

Subtask 4（55分）： $K \leq 400\,000$

- 7分： $2 \leq N \leq 7$
- 20分： $2 \leq N \leq 50$
- 25~55分： A 是你的操作次数， S 是参考答案的操作次数，你的得分为（或参考上表）：

$$25 + \left\lfloor 10 \cdot \tan(\arctan(3)) \cdot \max\left(0, \min\left(1, 1 - \frac{A - S}{400\,000 - S}\right)\right)\right\rfloor$$

maximum A	possible score
68 840	55
68 841	54
80 000	51
90 000	48
100 000	46
200 000	34
300 000	28
400 000	25

总结

总结

- 包括欧洲各地信息学竞赛在内的许多国际比赛有不少形式新颖、思维方面要求较高、但不涉及高深的数据结构和算法的题目。这些题目和解题方法，可供中国选手参加各类国际比赛前练习和参考，也可为拓展NOI系列活动的科技树增添更多思路。
- 部分欧洲国家在人口基数少的情况下，仍然产生了一定数量的优秀选手，他们的培养模式和各轮、各阶段的题目值得我们参考和取长补短，特别是吸纳一些他们鼓励入门选手参与的方法。
- 最后，感谢同学们参与本次冬令营课程，并欢迎提出你们的想法和建议：)