



中国计算机学会
China Computer Federation



欧洲信息学竞赛题目选讲

NOI2023 冬令营
钟诚



中国计算机学会
China Computer Federation



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





- 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（EGOI国家队选拔赛）
 - 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：决赛（IOI、CEOI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
- 有包括作业、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



中国计算机学会
China Computer Federation



题目选讲

部分题目可能有多种解法，欢迎同学们踊跃发言！

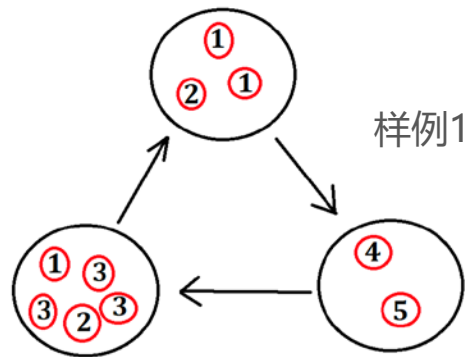


三角形演讲 (瑞典程序设计奥林匹克 PO-Final 2022)

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



中国计算机学会
China Computer Federation



三角形演讲（瑞典程序设计奥林匹克 PO-Final 2022）

Subtask 1: 如果所有 $A[i]$ 都相同，那么我们就希望最小的组尽可能大，只要分为3个尽可能大小一样的组，看是否符合要求即可。





中国计算机学会
China Computer Federation



三角形演讲 (瑞典程序设计奥林匹克 PO-Final 2022)

Subtask 2: 如果 $N \leq 10$, 那么可以枚举所有的 3^N 种分组的方法。





三角形演讲 (瑞典程序设计奥林匹克 PO-Final 2022)

Subtask 3: 如果 $A[i] \leq 3$, 那么当 N 至少是9的时候, 可以分为3组, 否则按Subtask 2的方法, 枚举所有的 3^N 种分组的方法。





三角形演讲 (瑞典程序设计奥林匹克 PO-Final 2022)

Subtask 4: 将 $A[i]$ 数组排序, 那么3个分组将是连续的3段。否则, 如果 $A[i] < A[j] < A[k]$, 且 i 和 k 属于同一个小组, j 属于另一个小组, 那么可以将 i 和 j 换一换。例如:



因此, 可以用 $O(N^2)$ 的算法枚举所有可能的分组。



三角形演讲 (瑞典程序设计奥林匹克 PO-Final 2022)

Subtask 5: 不妨设第3组里的数最大。如果有满足要求的分组, 那么一定存在一个方案, 使得第2组的大小恰好是第1组的最大元素。例如:



因此, 可以用 $O(N)$ 的算法枚举所有可能的分组。



三角形演讲 (瑞典程序设计奥林匹克 PO-Final 2022)

Legend: ✓ First to solve ✓ Full score 🟡 Partial score ✗ Attempted ⌚ Pending judgement

| TEAM | A (100) | B (100) | C (100) | D (100) | E (100) | F (100) | SCORE | TIME |
|---------------------------|------------------|-------------------|-------------------|-------------------|------------------|-----------------|-------|------|
| 1 Victor Vatn | ✓ 100 1 try | ✓ 100 1 try | ✓ 100 2 tries | ✓ 100 2 tries | ✓ 68 16 tries | 🟡 33 2 tries | 501 | 281 |
| 2 Olle Lapidus | ✓ 100 1 try | ✓ 100 1 try | ✓ 100 3 tries | ✓ 100 36 tries | ✓ 68 10 tries | ✗ 0 1 try | 468 | 290 |
| 3 Adam Amanbaev | ✓ 100 1 try | ✓ 100 3 tries | ✓ 100 2 tries | ✓ 100 3 tries | 🟡 30 3 tries | | 430 | 269 |
| 4 Jakob Puhl | ✓ 100 2 tries | ✓ 100 2 tries | ✓ 100 11 tries | 🟡 41 15 tries | ✓ 64 1 try | | 405 | 287 |
| 5 Anton Magnusson | ✓ 100 2 tries | ✓ 100 3 tries | ✓ 100 1 try | ✓ 100 2 tries | ✗ 0 2 tries | | 400 | 247 |
| 6 Oskar Paulsson | ✓ 100 2 tries | ✓ 100 2 tries | ✓ 100 2 tries | ✓ 100 21 tries | | | 400 | 285 |
| 7 Alexander Wahlsten | ✓ 100 1 try | ✓ 100 11 tries | ✓ 100 1 try | 🟡 14 5 tries | ✓ 84 3 tries | | 398 | 269 |
| 8 Erik Hedin | ✓ 100 1 try | ✓ 100 2 tries | ✓ 100 2 tries | 🟡 64 4 tries | 🟡 18 2 tries | | 382 | 279 |
| 9 Elias Lundell | ✓ 100 2 tries | ✓ 100 2 tries | ✓ 100 1 try | 🟡 14 5 tries | 🟡 18 1 try | | 332 | 298 |
| 10 William Kraft | ✓ 100 1 try | ✓ 100 3 tries | ✓ 100 1 try | 🟡 14 9 tries | | | 314 | 269 |
| 11 Joel Niemelä | ✓ 100 1 try | ✓ 100 2 tries | 🟡 50 1 try | 🟡 30 6 tries | | | 280 | 293 |
| 12 Alexander Konstantinov | ✓ 100 3 tries | ✓ 100 2 tries | ✗ 10 3 tries | 🟡 14 4 tries | 🟡 54 7 tries | | 278 | 272 |
| 13 Adrian Wireklint | ✓ 100 1 try | 🟡 60 9 tries | ✓ 100 6 tries | 🟡 14 6 tries | ✗ 0 2 tries | | 274 | 282 |
| 14 Nils Olsson | ✓ 100 1 try | ✓ 100 7 tries | 🟡 50 10 tries | 🟡 14 7 tries | | | 264 | 203 |
| 15 Martin Larsson | ✓ 100 1 try | 🟡 60 3 tries | ✓ 100 7 tries | ✗ 0 4 tries | | | 260 | 183 |



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);
```

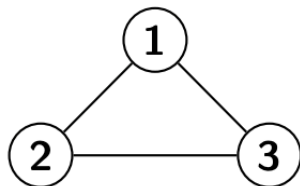
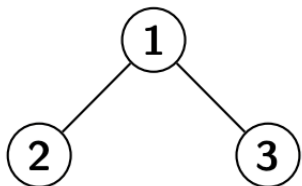
表示询问第 a 号设计中，引脚 i 和引脚 j 是否直接或间接相连。该测试器返回一个整数，如果相连，则直接返回 a 。如果不相连，则将创建一个新的设计，该设计是第 a 号设计的基础上加上 i 与 j 连的边，设计的编号为下一个可用的编号，并返回这个编号。无论如何，不会修改原先的第 a 号设计。



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 |
|---------------------------------------|-------------------------------|--|
| | <code>ToyDesign(4, 20)</code> | $n = 4, \text{max_ops} = 20$ |
| <code>Connected(0, 1, 2)</code> | 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。 |
| <code>Connected(1, 3, 2)</code> | 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。 |
| <code>Connected(0, 3, 4)</code> | Returns 0. | 在第0号设计中, 引脚3和引脚4直接或间接相连。 |
| <code>DescribeDesign({{3, 4}})</code> | - | 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



中国计算机学会
China Computer Federation



玩具设计 (欧洲女生信息学奥林匹克 EGOI 2022)

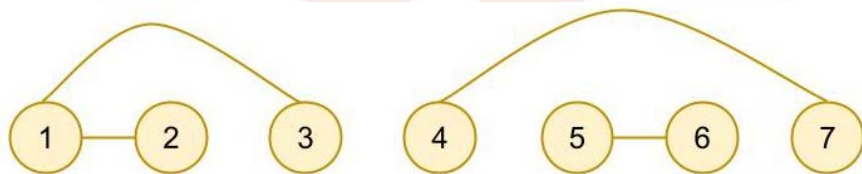
Subtask 1: 枚举所有的引脚对, 查看是否相连, 即每次只调用`Connected(0, i, j)`。





玩具设计 (欧洲女生信息学奥林匹克 EGOI 2022)

Subtask 4: 我们从1至n逐一处理每个引脚。假设处理完k个引脚后，有u个连通块。那么当我们处理引脚k+1时，希望找到最小的i，使得前i个引脚合并起来后，与引脚k+1连通。例如在下图中，当k+1为5时，不存在这样的i；当k+1为6时，最小的i为5；当k+1为7时，最小的i为4。



换句话说，我们希望知道k+1在哪个连通块里。上述寻找过程可以通过二分法在 $\text{ceil}(\log_2 u + 1)$ 的询问次数里找到最小的i，或得出i不存在的结论。



玩具设计 (欧洲女生信息学奥林匹克 EGOI 2022)

至于前若干个引脚的合并，其实也不难维护：在二分查找时，如果存在这样的 i ，那么前 k 个引脚的合并等价于前 $k+1$ 个引脚的合并；如果不存在，那么也必然尝试过查询前 k 个引脚的合并是否与引脚 $k+1$ 连通，此时便生成了一个新的设计，即为前 $k+1$ 个引脚的合并。

因此，总共调用测试器 $\log_2 1 + \log_2 2 + \dots + \log_2 n$ 次，即可获得一个与第0号设计相同的设计。



中国计算机学会
China Computer Federation



玩具设计 (欧洲女生信息学奥林匹克 EGOI 2022)

| European Contestants | | | | All Contestants | | | | | | Online Contestants | | | | |
|----------------------|--------------------------|--------------|------|-----------------|----------|-------------------|----------|-------|-------------|--------------------|-----------|-------|-------|--------|
| Rank | First Name | Last Name | Team | SubsetMex | LegoWall | SocialEngineering | Tourists | Day.1 | Datacenters | Superpiece | JoyDesign | Cheat | Day.2 | Global |
| 1 | Ketevan | Tsimakuridze | | 100 | 100 | 100 | 100 | 400 | 77 | 61 | 100 | 53 | 291 | 691 |
| 2 | Duru | Özer | | 100 | 100 | 100 | 50 | 350 | 100 | 55 | 100 | 53 | 308 | 658 |
| 3 | Alisa | Potomkova | | 100 | 100 | 100 | 50 | 350 | 100 | 61 | 100 | 26 | 287 | 637 |
| 4 | Eliška | Macáková | | 100 | 100 | 100 | 50 | 350 | 100 | 100 | 45 | 26 | 271 | 621 |
| 5 | Daryna | Karpenko | | 100 | 100 | 30 | 50 | 280 | 100 | 61 | 100 | 71 | 332 | 612 |
| 5 | Alice | Tosel | | 100 | 70 | 100 | 75 | 345 | 77 | 61 | 100 | 29 | 267 | 612 |
| 7 | Jamie Jia Sin | Lim | | 100 | 44 | 100 | 50 | 294 | 100 | 100 | 10 | 100 | 310 | 604 |
| 8 | Olivia | Tennisberg | | 100 | 70 | 35 | 50 | 255 | 77 | 100 | 100 | 71 | 348 | 603 |
| 9 | Claire | Zhang | | 100 | 80 | 35 | 50 | 265 | 100 | 47 | 100 | 53 | 300 | 565 |
| 10 | Joanna | Suwaj | | 100 | 80 | 100 | 75 | 355 | 33 | 61 | 100 | 0 | 194 | 549 |
| 11 | Yulia | Tatarinova | | 100 | 44 | 75 | 0 | 219 | 100 | 100 | 100 | 29 | 329 | 548 |
| 12 | Jasmin | Studer | | 100 | 70 | 100 | 50 | 320 | 100 | 55 | 65 | 2 | 222 | 542 |
| 13 | Lior | Altahan | | 100 | 80 | 75 | 0 | 255 | 100 | 55 | 100 | 15 | 270 | 525 |
| 14 | Cynthia Naeli López | Estrada | | 100 | 74 | 20 | 50 | 244 | 77 | 47 | 100 | 53 | 277 | 521 |
| 15 | Lara | Semeš | | 100 | 44 | 50 | 25 | 219 | 77 | 94 | 100 | 11 | 282 | 501 |
| 16 | Fernanda Sarahy Mancilla | Núñez | | 100 | 44 | 20 | 25 | 189 | 100 | 42 | 100 | 53 | 295 | 484 |
| 17 | Alexandra-mihaela | Nicola | | 100 | 26 | 35 | 50 | 211 | 100 | 47 | 100 | 17 | 264 | 475 |
| 18 | Iulia | Slănină | | 100 | 70 | 15 | 50 | 235 | 100 | 55 | 10 | 53 | 218 | 453 |
| 19 | Katherine | Li | | 100 | 80 | 30 | 50 | 260 | 100 | 61 | 10 | 11 | 182 | 442 |
| 20 | Julia | Kędziora | | 100 | 80 | 35 | 50 | 265 | 100 | 61 | 10 | 0 | 171 | 436 |
| 21 | Yael | Ginzburg | | 100 | 44 | 30 | 50 | 224 | 77 | 49 | 80 | 2 | 208 | 432 |
| 22 | Paulina | Želežnik | | 100 | 44 | 35 | 50 | 229 | 77 | 61 | 10 | 53 | 201 | 430 |
| 23 | Fidan | Huseynova | | 100 | 26 | 20 | 25 | 171 | 77 | 55 | 100 | 26 | 258 | 429 |
| 24 | Anni | Tapionlinna | | 100 | 26 | 35 | 25 | 186 | 77 | 55 | 100 | 11 | 243 | 429 |



信息传递 (波罗的海信息学奥林匹克 BOI 2022)

你在一个潜水艇里，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 |

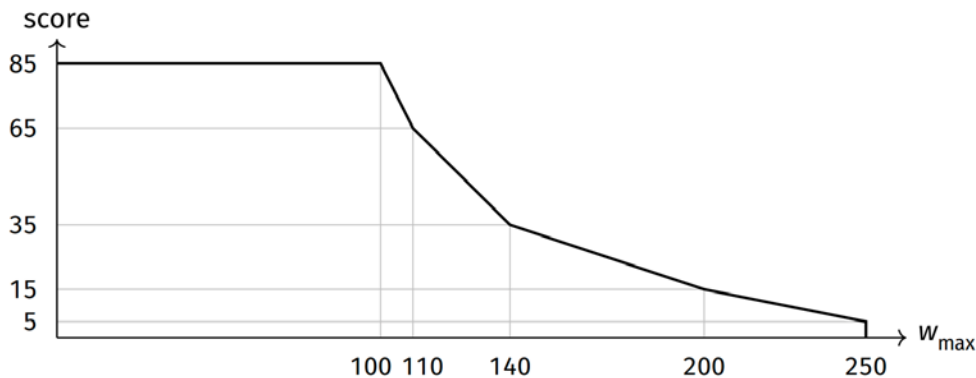


信息传递 (波罗的海信息学奥林匹克 BOI 2022)

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析法, 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 $\langle 2, 3 \rangle$
- 否则，如果有连续接收到两个0，同理X不可能是2，返回 $\langle 1, 3 \rangle$
- 否则，返回 $\langle 1, 2 \rangle$

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。



信息传递 (波罗的海信息学奥林匹克 BOI 2022)

Subtask 1: 当 $N = 3$ 时, 我们先看看, 有没有一种方法, 跟提示中的骗分算法一样, 不需要用到send函数的返回值。也就是说对于每个不同的 X , encode函数就固定send一串信息 $A[X]$ 。我们先随便找一组 $A[X]$ 看看, 比如 X 的二进制表示:

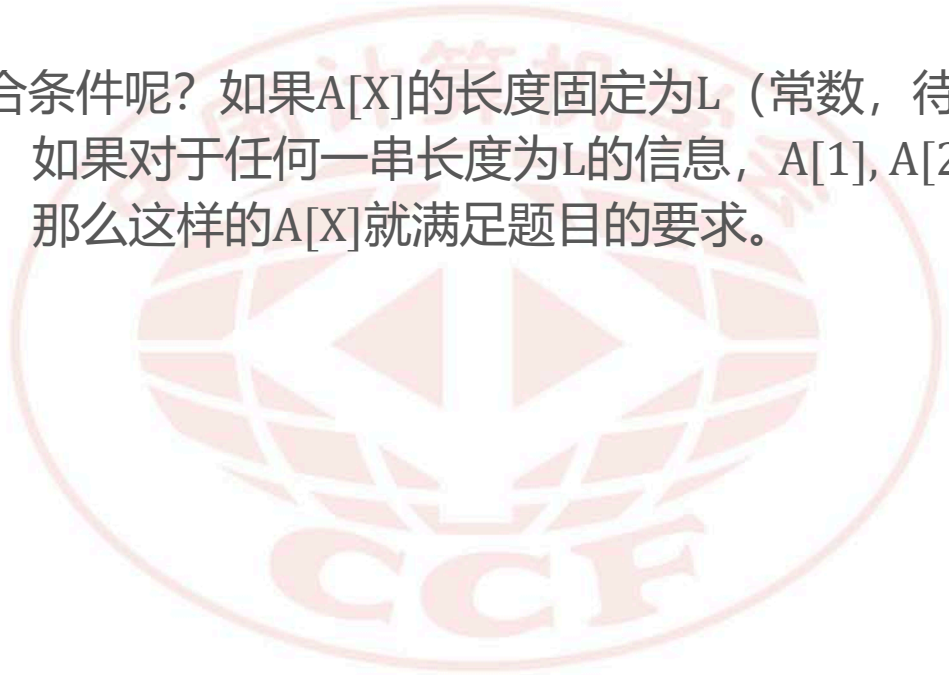
| | | | |
|--------|----|----|----|
| X | 1 | 2 | 3 |
| $A[X]$ | 00 | 01 | 10 |

我们发现这是不行的。如果decode函数收到了11, 那么可以确定 X 不等于1, 于是我们返回 $\langle 2, 3 \rangle$ 即可。但如果decode函数收到的是00就懵了, 因为 X 可以等于1, 2, 3中的任何一个值。



信息传递 (波罗的海信息学奥林匹克 BOI 2022)

那怎么样的 $A[X]$ 符合条件呢？如果 $A[X]$ 的长度固定为 L （常数，待确定），根据上述的尝试，我们发现，如果对于任何一串长度为 L 的信息， $A[1], A[2], A[3]$ 中至多有两个可以被传递成它，那么这样的 $A[X]$ 就满足题目的要求。





信息传递 (波罗的海信息学奥林匹克 BOI 2022)

那怎么样的 $A[X]$ 符合条件呢？如果 $A[X]$ 的长度固定为 L （常数，待确定），根据上述的尝试，我们发现，如果对于任何一串长度为 L 的信息， $A[1], A[2], A[3]$ 中至多有两个可以被传递成它，那么这样的 $A[X]$ 就满足题目的要求。

我们可以另写一个小程序从小到大枚举 L ，然后枚举所有可能的 $A[X]$ ，或者手算得出，当 $L = 4$ 时，下述 $A[X]$ 满足要求：

| | | | |
|--------|------|------|------|
| X | 1 | 2 | 3 |
| $A[X]$ | 0000 | 0110 | 1111 |

比如，decode收到了0010，那么我们可以确定 X 不等于3，如果收到了1101，可以确定 X 不等于1，等等。可以证明，对于任何一个长度为4的01串， $A[X]$ 中至少有一个不可能传递成它。



信息传递 (波罗的海信息学奥林匹克 BOI 2022)

Subtask 2 部分分解法：我们可以巧妙地利用Subtask 1的结论，对 $1 \sim N$ 进行三分。我们把 $T_0 = \{1, 2, 3, \dots, N\}$ 尽量均匀地分为三份 S_1, S_2, S_3 。如果 $X \in S_i$ ，那么我们就用4个比特传递 i 这一信息，助理收到后，虽然无法确定 i 准确的值，但至少可以排除掉1, 2, 3中的一个值。于是，可以确定 $X \in S_i \cup S_j$ （记 $T_1 = S_i \cup S_j$ ）。然后，我们可以再对 T_1 进行三分，重复这一过程。

考虑到 $10^9 * (2/3)^{49} \approx 2$ ，所以大概要进行50次三分即可将最终的结果确定在2个数的范围里，也就时大概需要200次查询，这样的话这题可以得30分（包括Subtask 1的15分）。



信息传递 (波罗的海信息学奥林匹克 BOI 2022)

在此基础上，还有一些优化的方法。例如，我们也可以考虑四分，因为在Subtask 1中，如果 $N = 4$ 的话，我们可以找到下述的 $A[X]$ ：

| X | 1 | 2 | 3 | 4 |
|--------|------|------|------|------|
| $A[X]$ | 0000 | 0110 | 1001 | 1111 |

这样的话，我们每次把T集合四分，四个子集里排除掉两个，就更快了。这样的话这题大概可以得75分。

如果分的份数更多，那么L也会更大。经过计算和实验，我们发现四分的效果最好。



信息传递 (波罗的海信息学奥林匹克 BOI 2022)

Subtask 2 满分解法：在之前的解法中，我们都没有用到send函数的返回值。在这里，我们用动态规划的思想，并把收到的信息是否有误作为动态规划状态的一个维度。

与上一解法一样，我们用 T_i 表示当前 x 所有可能的值的集合，并把它划分为两块： T_i^{corr} 和 T_i^{wrong} ，分别表示当前收到的（1比特）信息正确和错误的情况下， x 所有可能的值的集合。我们设计的encode算法如下：如果 $x \in T_i^{\text{corr}}/2 \cup T_i^{\text{wrong}}/2$ ，则在第 $i+1$ 次send(1)，否则send(0)；其中我们用 $T/2$ 表示 T 集合的一半元素组成的集合，方便起见不妨取较小的一半数。



信息传递 (波罗的海信息学奥林匹克 BOI 2022)

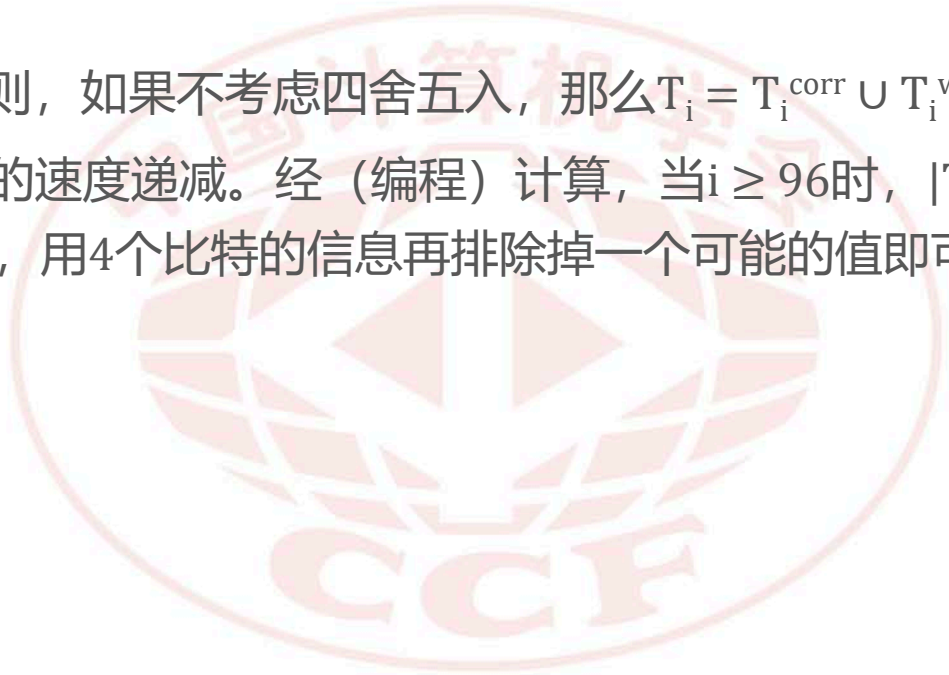
例如, $N = 100$, $X = 37$ 时, 可能有以下操作:

| i | T_i^{corr} | T_i^{wrong} | $T_i^{\text{corr}}/2 \cup T_i^{\text{wrong}}/2$ | 操作 | 返回值 |
|-------|--------------------------------------|--------------------------------------|---|---------|-----|
| 0 | $\{1 \sim 100\}$ | \emptyset | $\{1 \sim 50\}$ | / | / |
| 1 | $\{1 \sim 50\}$ | $\{51 \sim 100\}$ | $\{1 \sim 25\} \cup \{51 \sim 75\}$ | send(1) | 1 |
| 2 | $\{1 \sim 25\} \cup \{51 \sim 75\}$ | $\{26 \sim 50\}$ | $\{1 \sim 38\}$ | send(0) | 1 |
| 3 | $\{1 \sim 38\}$ | $\{51 \sim 75\}$ | $\{1 \sim 19\} \cup \{51 \sim 63\}$ | send(1) | 1 |
| 4 | $\{20 \sim 38\} \cup \{64 \sim 75\}$ | $\{1 \sim 19\}$ | $\{1 \sim 10\} \cup \{20 \sim 35\}$ | send(0) | 0 |
| 5 | $\{1 \sim 10\} \cup \{20 \sim 35\}$ | $\{36 \sim 38\} \cup \{64 \sim 75\}$ | $\{1 \sim 10\} \cup \{20 \sim 22\} \cup \{36 \sim 38\} \cup \{64 \sim 68\}$ | send(0) | 1 |
| | | | | | |



信息传递 (波罗的海信息学奥林匹克 BOI 2022)

根据这样的操作规则，如果不考虑四舍五入，那么 $T_i = T_i^{\text{corr}} \cup T_i^{\text{wrong}}$ 的元素个数将以 $n, \frac{3}{4}n, \frac{5}{8}n, \frac{13}{32}n, \dots$ 的速度递减。经（编程）计算，当 $i \geq 96$ 时， $|T_i| \leq 3$ 。此时用 Subtask 1 中的方法，用4个比特的信息再排除掉一个可能的值即可，因此共需要传递100次信息。





信息传递 (波罗的海信息学奥林匹克 BOI 2022)

这个算法是怎么想到的呢？初始时，我们可以考虑每次有一个集合 P_i ，当 $X \in P_i$ 时，在第 $i+1$ 次send(1)，否则send(0)。那么：

- 当返回值为1时， $T_{i+1}^{\text{corr}} = P_i \cap (T_i^{\text{corr}} \cup T_i^{\text{wrong}})$, $T_{i+1}^{\text{wrong}} = T_i^{\text{corr}} - P_i$
- 当返回值为0时， $T_{i+1}^{\text{corr}} = (T_i^{\text{corr}} \cup T_i^{\text{wrong}}) - P_i$, $T_{i+1}^{\text{wrong}} = P_i \cap T_i^{\text{corr}}$

为了让 $T_i = T_i^{\text{corr}} \cup T_i^{\text{wrong}}$ 的元素个数尽可能稳定地减少，且不受每次具体返回值的影
响（我们总要假设评测机是非常聪明的），一种可行的做法就是让 $T_i^{\text{corr}} \cap P_i = T_i^{\text{corr}} / 2$, $T_i^{\text{wrong}} \cap P_i = T_i^{\text{wrong}} / 2$ 。



中国计算机学会
China Computer Federation



信息传递 (波罗的海信息学奥林匹克 BOI 2022)

为什么本题仅要求最后给出两个可能的值，而不是精确到一个呢？





信息传递 (波罗的海信息学奥林匹克 BOI 2022)

为什么本题仅要求最后给出两个可能的值，而不是精确到一个呢？当 $N = 2$ 的时候，只要评测机足够聪明，可以对`encode(2, 1)`和`encode(2, 2)`这两个函数每次调用`send`后的返回值完全相同。

具体方法是：轮流返回这两个函数`send`的信息，这样能保证任何连续两次发送时，最多有一次是错误的，且两个函数`send`的返回值完全相同。这样，`decode`函数就一定无法区分。



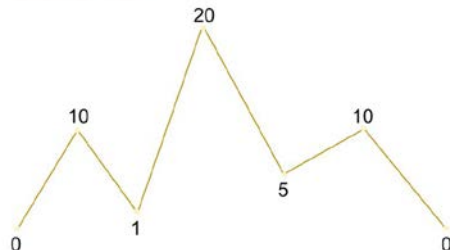
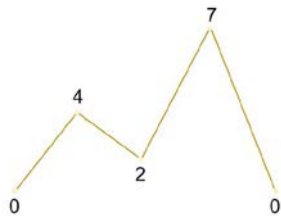
信息传递 (波罗的海信息学奥林匹克 BOI 2022)

| # | Contestant | Team | art | events | vault | communication | island | passes | Score |
|----|-------------------------|-----------|-----|--------|-------|---------------|--------|--------|-------|
| 1 | Antoni Buraczewski | Poland | 100 | 55 | 80 | 72 | 100 | 100 | 507 |
| 2 | Matiss Kristiņš | Latvia | 100 | 100 | 80 | 0 | 100 | 100 | 480 |
| 3 | Oleh Naver | Ukraine | 100 | 100 | 25 | 15 | 100 | 100 | 440 |
| 4 | Roman Yanushevskiy | Ukraine | 100 | 70 | 40 | 15 | 100 | 100 | 425 |
| 5 | Henrik Aalto | Finland | 100 | 100 | 5 | 15 | 100 | 100 | 420 |
| | Elazar Koren | Israel B | 100 | 100 | 20 | 15 | 100 | 60 | 395 |
| 6 | Andrii Fylypiuk | Ukraine | 100 | 100 | 60 | 0 | 20 | 100 | 380 |
| 6 | Vladyslav Denysiuk | Ukraine | 100 | 100 | 20 | 0 | 100 | 60 | 380 |
| 8 | Lucas Schwebler | Germany | 20 | 100 | 0 | 74 | 100 | 60 | 354 |
| 9 | Daniel Weber | Israel | 100 | 100 | 20 | 15 | 100 | 5 | 340 |
| 10 | Marko Tsengov | Estonia | 100 | 100 | 5 | 0 | 100 | 30 | 335 |
| 10 | Alon Tanay | Israel | 100 | 70 | 5 | 0 | 100 | 60 | 335 |
| 12 | Aldas Lenkšas | Lithuania | 100 | 30 | 0 | 0 | 100 | 100 | 330 |
| 12 | Hubert Wasilewski | Poland | 100 | 30 | 0 | 0 | 100 | 100 | 330 |
| 12 | Ansis Gustavs Andersons | Latvia | 100 | 30 | 0 | 0 | 100 | 100 | 330 |
| 15 | Victor Vatn | Sweden | 100 | 55 | 5 | 15 | 100 | 0 | 275 |
| 16 | Krzysztof Olejnik | Poland | 70 | 40 | 0 | 0 | 100 | 60 | 270 |
| 16 | Niklas Leinert | Germany | 100 | 10 | 0 | 0 | 100 | 60 | 270 |
| 18 | Lorenzo Ferrari | Denmark | 100 | 10 | 20 | 0 | 35 | 100 | 265 |
| 19 | Yahli Hecht | Israel | 100 | 30 | 5 | 0 | 65 | 60 | 260 |
| 20 | Kregor Ööbik | Estonia | 100 | 60 | 0 | 0 | 35 | 60 | 255 |
| 21 | Oleksandr Kozlovskiy | Ukraine | 100 | 10 | 5 | 0 | 100 | 30 | 245 |
| | Eitan Elbaum | Israel B | 100 | 100 | 5 | 15 | 20 | 5 | 245 |



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

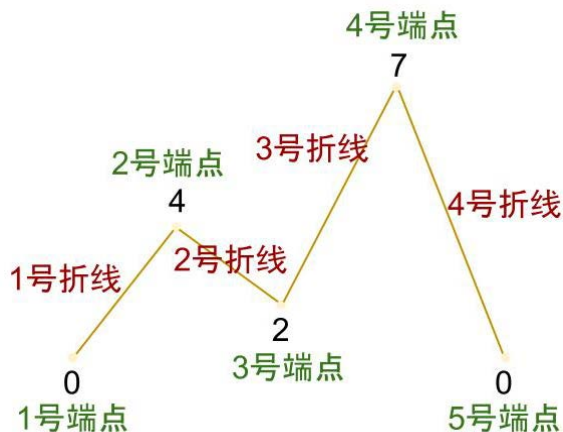
Subtask 3 (50分)： $3 \leq N \leq 5\,000$ ，所有中间端点的高度介于1和 10^6 之间



登山者 (罗马尼亚信息学大师赛 RMI 2018)

显然，对于0 1 3 7 5 2 8 5 0这样的折线，在本题中与0 7 2 8 0这样的折线是等价的，因此下面我们不妨设折线里的高度一定是高低交错的。我们把折线段中的端点用1, 2, ..., n编号，并记他们的高度为 h_1, h_2, \dots, h_n 。我们把第i个端点和第i+1个端点中的那部分折线，称为第i号折线（它也包括两个端点）。

对于任何一个最优解，爬山方向（向上/向下）改变的时刻，Alice和Bob中至少一人正处于某个端点上。否则，如果他们都在半山腰（各自所在折线的中间），他们只能都往回走，是毫无意义的浪费体力。





登山者 (罗马尼亚信息学大师赛 RMI 2018)

因此，我们可以定义这样的登山状态 (x, y) ，表示其中一人（Alice或Bob）在 x 号端点，另一人在 y 号折线上。例如在下图中，一个最优解的登山状态是这样依次进行的： $(1, 7) \rightarrow (2, 6) \rightarrow (3, 6) \rightarrow (6, 3) \rightarrow (5, 3) \rightarrow (4, 4)$ 。

其中， $(6, 3)$ 的状态在图中做了特别的标注。注意，并非所有的 (x, y) 都存在，例如在上图中 $(6, 2)$ 就不存在，因为6号端点的高度为5，而2号折线中不存在高度为5的位置。

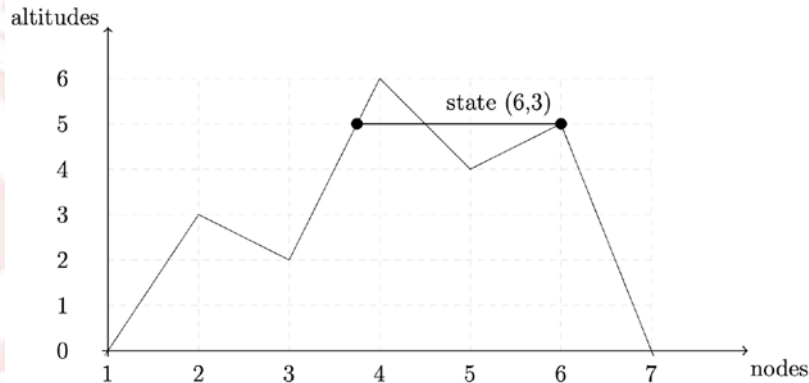


Figure 1: A sample landscape.



登山者 (罗马尼亚信息学大师赛 RMI 2018)

于是，我们可以构造一个有向图，其中

- 图的顶点为登山状态
- 图的边为登山状态之间的转移
- 边的权值为两个状态之间高度之差，即状态转移时消耗的体力值
- 起点是登山状态(1, n)
- 任何一个登山状态(x, x)都可以作为终点

对于每个顶点（登山状态），它的度是

- 2, 如果 $h_y < h_x < h_{y+1}$ 。因为此时在x号端点的登山者可以选择向左或者向右，而在y号折线的登山者只有一个选择。
- 4, 如果 $h_x = h_{y+1}$ 或者 $h_x = h_y$ 。因为此时两个登山者都可以选择向左或者向右。
- 1, 对于登山状态(1, n)。
- 1, 对于登山状态(x, x)，因为两个登山者只能回退到他们来的地方。



中国计算机学会
China Computer Federation



登山者 (罗马尼亚信息学大师赛 RMI 2018)

对于Subtask 1, 由于不存在度为4的顶点, 我们直接从一个登山状态走到下一个有边相连的登山状态, 直到走到任何一个 (x, x) 即可。

而对于Subtask 3, 我们可以用Dijkstra算法找到从 $(1, n)$ 到任何一个 (x, x) 的最短路。





中国计算机学会
China Computer Federation



登山者 (罗马尼亚信息学大师赛 RMI 2018)

| | | | | RMI Final | | | | | | | | | |
|-----------------|---------|-------------------|-------------|-----------|----------|-------------|-------|--------|----------|-----|-------|--------|--|
| Username | Country | First Name | Last Name | climbers | password | traffickers | Day 1 | colors | squirrel | w | Day 2 | Global | |
| gainullin-7bf | RUS_K | Ildar | Gainullin | 100 | 100 | 100 | 300 | 100 | 100 | 100 | 300 | 600 | |
| dimitrov-40a | BG | Radoslav Stoyanov | Dimitrov | 90 | 100 | 100 | 290 | 100 | 80 | 100 | 280 | 570 | |
| morozov-53c | RUS_NW | Aleksandr | Morozov | 100 | 100 | 100 | 300 | 100 | 100 | 50 | 250 | 550 | |
| moroianu-64b | RO | Theodor-Pierre | Moroianu | 100 | 80 | 100 | 280 | 100 | 45 | 100 | 245 | 525 | |
| savkin-45d | RUS_M | Semen | Savkin | 100 | 100 | 100 | 300 | 100 | 75 | 50 | 225 | 525 | |
| lifar-cbe | RUS_M | Egor | Lifar | 5 | 100 | 100 | 205 | 78 | 100 | 100 | 278 | 483 | |
| musat-4e2 | RO_V1 | Tiberiu | Musat | 100 | 100 | 100 | 300 | 47 | 100 | 20 | 167 | 467 | |
| krastev-fa9 | BG | Aleksandar Tonev | Krastev | 55 | 80 | 100 | 235 | 78 | 100 | 0 | 178 | 413 | |
| sitaru-6d0 | RO_2 | Bogdan | Sitaru | 100 | 100 | 100 | 300 | 62 | 25 | 25 | 112 | 412 | |
| romanov-385 | RUS_M | Vladimir | Romanov | 30 | 100 | 100 | 230 | 100 | 15 | 30 | 145 | 375 | |
| odintsov-1e2 | RUS_NW | Andrei | Odintsov | 0 | 100 | 100 | 200 | 47 | 100 | 20 | 167 | 367 | |
| pop-0b1 | RO_2 | Bogdan Petru | Pop | 100 | 50 | 100 | 250 | 47 | 50 | 20 | 117 | 367 | |
| todorov-797 | BG_Y | Andon Ivaylov | Todorov | 30 | 100 | 45 | 175 | 22 | 100 | 0 | 122 | 297 | |
| nyagolov-94b | BG | Petar Invelinov | Nyagolov | 5 | 100 | 100 | 205 | 0 | 25 | 55 | 80 | 285 | |
| pipis-339 | GR | Charilaos | Pipis | 5 | 45 | 100 | 150 | 47 | 0 | 85 | 132 | 282 | |
| dimitrov-df3 | BG_SH | Yordan Dimitrov | Dimitrov | 5 | 55 | 100 | 160 | 100 | 0 | 15 | 115 | 275 | |
| kopchev-fdb | BG | Martin Daniel | Kopchev | 5 | 50 | 15 | 70 | 62 | 100 | 30 | 192 | 262 | |
| shekhovtsov-376 | RUS_M | Alexander | Shekhovtsov | 60 | 45 | 15 | 120 | 100 | 10 | 30 | 140 | 260 | |
| dicu-d79 | RO | Adrian Emanuel | Dicu | 5 | 100 | 100 | 205 | 0 | 45 | 0 | 45 | 250 | |
| petrescu-163 | RO_V1 | Alexandru | Petrescu | 0 | 50 | 100 | 150 | 47 | 30 | 20 | 97 | 247 | |
| coman-5be | RO_V1 | Andrei | Coman | 0 | 50 | 100 | 150 | 7 | 35 | 50 | 92 | 242 | |
| grigorev-fb7 | RUS_C | Savelli | Grigorev | 55 | 80 | 15 | 150 | 47 | 25 | 20 | 92 | 242 | |
| gospodinov-57f | BG_Y | Antani Antonov | Gospodinov | 0 | 80 | 55 | 135 | 62 | 15 | 25 | 102 | 237 | |
| ismagilov-6cb | RUS_K | Azat | Ismagilov | 0 | 100 | 15 | 115 | 62 | 50 | 10 | 122 | 237 | |
| pipis-89f | GR | Evangelos | Pipis | 0 | 100 | 15 | 115 | 47 | 50 | 20 | 117 | 232 | |
| puzic-ef7 | SRB | Mladen | Puzic | 0 | 50 | 15 | 65 | 47 | 100 | 20 | 167 | 232 | |
| tinca-46b | RO_V1 | Matei | Tinca | 5 | 100 | 100 | 205 | 0 | 0 | 20 | 20 | 225 | |
| popovici-b14 | RO_V2 | Robert | Popovici | 5 | 55 | 60 | 120 | 78 | 0 | 20 | 98 | 218 | |
| bortolin-8f5 | IT | Alessandro | Bortolin | 35 | 100 | 0 | 135 | 47 | 15 | 20 | 82 | 217 | |
| karagyozev-ef3 | BG_V1 | Ivo | Karagyozev | 40 | 50 | 60 | 150 | 22 | 25 | 20 | 67 | 217 | |
| tudose-80d | RO | Maria-Alexa | Tudose | 5 | 80 | 80 | 165 | 7 | 15 | 30 | 52 | 217 | |
| georgescu-a12 | RO_B | Laura Ioana | Georgescu | 5 | 40 | 100 | 145 | 47 | 10 | 0 | 57 | 202 | |
| kozhuharov-ce5 | BG_P | Viktor | Kozhuharov | 55 | 50 | 0 | 105 | 47 | 25 | 20 | 92 | 197 | |
| budnikov-bde | RUS_C | Mikhail | Budnikov | 65 | 15 | 55 | 135 | 22 | 15 | 20 | 57 | 192 | |
| claudiu-c84 | MD | Babin | Claudiu | 0 | 80 | 55 | 135 | 22 | 15 | 20 | 57 | 192 | |
| szczepanski-1c7 | PL_2 | Kacper | Szczepanski | 0 | 30 | 45 | 75 | 69 | 15 | 25 | 109 | 184 | |
| korsic-c7a | SRB | Lazar | Korsic | 65 | 100 | 15 | 180 | 0 | 0 | 0 | 0 | 180 | |



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。

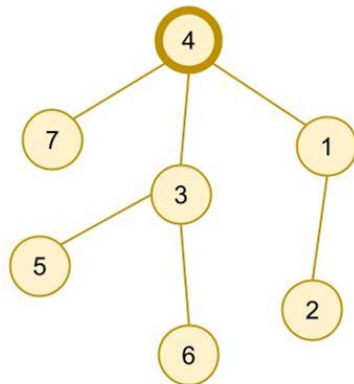
在第一次询问中，节点5、6的最近公共祖先为节点3，并不是5、6中的一个，因此交互库回答NO。

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。



寻找树根 (欧洲初中信息学奥林匹克 EJOI 2022)

Subtask 2:

在我们不知道哪个节点是根的情况下，为了便于后续讨论，我们统一把度为1的节点称呼为叶节点，度大于1的节点称呼为中间节点。在这样的称呼模式下，最后找到的根既可能是叶节点，也可能是中间节点。

我们询问所有的节点对。如果有一个节点 $r_{\text{candidate}}$ ，它对于所有的询问 $\{r_{\text{candidate}}, v\}$ ($1 \leq v \leq n, v \neq r_{\text{candidate}}$)，交互库都会回答YES，那么我们称它为候选根节点。

显然，根节点一定是候选根节点。如果只有一个节点是候选根节点，那么它就是根。那是否可能有其它节点是候选根节点呢？

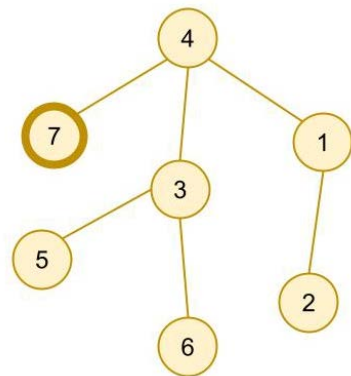


寻找树根 (欧洲初中信息学奥林匹克 EJOI 2022)

是否可能有其它节点是候选根节点呢？答案是有可能。例如在右图中，如果节点7是根节点，那么节点4、节点7都是候选根节点。下面我们来考察候选根节点的一些性质。

结论1：如果有多个节点是候选根节点，那么根一定是叶节点

证明：假设有一个非根节点 r_1 是候选根节点。如果根 r 是中间节点，那么它就有多个子节点。那么，就一定存在一个 v ，它和 r_1 在不同的子树上，意味着 r_1 和 v 的最近公共祖先一定是 r ，矛盾。因此，这种情况下的根一定是叶节点。





寻找树根 (欧洲初中信息学奥林匹克 EJOI 2022)

结论2: 不是根的候选根节点一定是中间节点

证明: 假设有一个非根的叶子节点 r_1 是候选根节点。我们考虑另一个叶子节点 v (由于至少有一个节点与其它至少3个节点相连, 故至少有3个叶节点), 显然对于询问 $\{r_1, v\}$ 的回答一定是NO, 因为它们都是非根的叶节点, 矛盾。

综合以上两个结论, 如果仅有一个候选根节点, 那么它就是根; 如果有多个候选根节点, 那么根一定是叶节点中唯一的一个候选根节点。这样, 我们在 $n(n-1)/2$ 次询问之后, 就能找到根节点。



寻找树根 (欧洲初中信息学奥林匹克 EJOI 2022)

Subtask 3 部分分解法:

设根为 r 。我们先询问一次所有叶节点组成的集合。显然, 它们的最近公共祖先是 r , 只要按 r 是否为叶节点分情况讨论即可得出此结论。于是通过这次询问, 我们就知道了 r 是否为叶节点。

第一种情况, r 是中间节点。我们每次询问部分中间节点和全部叶节点组成的集合, 它们的最近公共祖先一定是 r , 这样我们通过交互库的回答, 就知道 r 是否在该集合内。于是, 我们可以通过二分法, 在所有的中间节点里找到 r 。



寻找树根 (欧洲初中信息学奥林匹克 EJOI 2022)

第二种情况, r 是叶节点。我们每次询问部分 (≥ 2) 叶节点组成的集合。当且仅当 r 在集合内时, 交互库会回答 YES。于是, 我们也可以通过二分法, 在所有的叶节点里寻找 r , 并把范围缩小到 r_1, r_2 两个节点。随后, 我们另找一个叶节点 r_3 , 并询问 $\{r_2, r_3\}$, 就能判断出 r_2 是否为根。

这样, 我们可以通过 $1 + 9 = 10$ 次询问, 找到树根。



寻找树根 (欧洲初中信息学奥林匹克 EJOI 2022)

Subtask 3 满分解法:

我们的上一个解法在二分法开始前, 先检查了根是叶节点还是中间节点。那么是否有可能在不知道根是哪种节点的情况下进行二分呢?

我们换一种二分的方法。先对所有的节点按度从小到大排序(事实上, 只需要把叶节点排在前面, 中间节点排在后面即可)。然后, 我们用二分法找到最小的 k , 使得询问前 k 个节点组成的集合时, 交互库回答YES。

这样, 每次询问的集合要么全部是叶节点, 要么是所有的叶节点和部分中间节点。无论是哪种情况, 根据上一个解法中的讨论, 我们可以知道, 当且仅当 r 在集合中时, 交互库会回答YES。



寻找树根 (欧洲初中信息学奥林匹克 EJOI 2022)

唯一的特殊情况是我们二分到剩下前2个节点 r_1, r_2 时, 发现交互库还是回答YES, 那这时候我们就可像上一个解法一样, 另找一个叶节点 r_3 , 并询问 $\{r_2, r_3\}$, 就能判断出 r_2 是否为根。

这样, 我们可以通过9次询问, 找到树根。



寻找树根 (欧洲初中信息学奥林匹克 EJOI 2022)

| Rank | Name | Country | A | B | C | Day 1 | A | B | C | Day 2 | Total |
|-------|-------------------------|-----------------|-----|-----|----|-------|-----|-----|----|-------|-------|
| 1 | Oscar Fischler | France | 100 | 100 | 27 | 227 | 100 | 100 | 21 | 221 | 448 |
| 2 | Ștefan Cătălin Savu | Romania | 100 | 100 | 23 | 223 | 100 | 37 | 60 | 197 | 420 |
| 3 | Alisa Potomkina | Ukraine | 100 | 28 | 17 | 145 | 100 | 100 | 13 | 213 | 358 |
| 4 | Mihai Valeriu Voicu | Romania | 100 | 83 | 27 | 210 | 33 | 100 | 13 | 146 | 356 |
| 5 | Gagik Gevorgyan | Armenia | 100 | 83 | 17 | 200 | 29 | 34 | 60 | 123 | 323 |
| 6 | Denys Tereshchenko | Ukraine | 55 | 83 | - | 138 | 44 | 100 | 30 | 174 | 312 |
| 7 | Stanisław Lada | Poland | 100 | 83 | 23 | 206 | 29 | 29 | 13 | 71 | 277 |
| 8 | Fabijan Cikač | Croatia | 100 | 83 | 17 | 200 | 29 | 9 | 13 | 51 | 251 |
| - | Huu-Tuan Nguyen | Viet Nam | 100 | 17 | 23 | 140 | 62 | 19 | 21 | 102 | 242 |
| 9 | Pál Czani | Hungary | 55 | 72 | 0 | 127 | 3 | 37 | 38 | 78 | 205 |
| 10-11 | Mircea Rebeniuc | Romania | 100 | 0 | 0 | 100 | 44 | 37 | 13 | 94 | 194 |
| 10-11 | Chenglin Shang | Romania | 100 | 0 | 10 | 110 | 62 | 9 | 13 | 84 | 194 |
| - | Ilnat Zharikhin | Ukraine | 100 | 17 | 10 | 127 | 9 | 37 | 13 | 59 | 186 |
| 12 | Efe Bilgin | Turkey | 30 | 34 | - | 64 | 9 | 100 | 13 | 122 | 186 |
| 13 | Damjan Davkov | North Macedonia | 0 | 76 | - | 76 | 62 | 13 | 30 | 105 | 181 |
| - | Jan Myszk | Poland | 55 | 27 | 0 | 82 | 18 | 57 | 13 | 88 | 170 |
| 14 | Fuad Garayev | Azerbaijan | 55 | 0 | 27 | 82 | 18 | 57 | 13 | 88 | 170 |
| 15 | Kipras Deksnys | Lithuania | 55 | 83 | - | 138 | 9 | 9 | 3 | 21 | 159 |
| 16 | Yersultan Abzhani | Kazakhstan | 0 | - | 27 | 27 | 0 | 100 | 30 | 130 | 157 |
| - | Abdulloshid Zhanabergen | Kazakhstan | 55 | 17 | 4 | 76 | 44 | 19 | 13 | 76 | 152 |
| - | Amirali Asgari | Iran | 100 | 27 | 4 | 131 | - | 14 | - | 14 | 145 |



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, ..., 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)

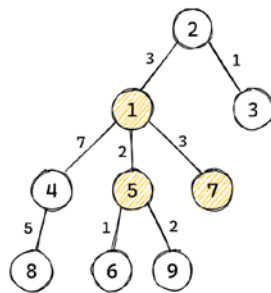
| 输入 | 输出 |
|---|------------------------------|
| 9 3 2 3 2 -1 2 1 1 5 1 4 5 9 4 5 5 7 3 -1 3 7 | |
| | 1 5 7 ? 1 5 ? 1 7 ! |
| 0 2 5 3 0 3 5 0 1 7 7 5 5 1 | |
| | 3 5 5 3 2 8 |

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。

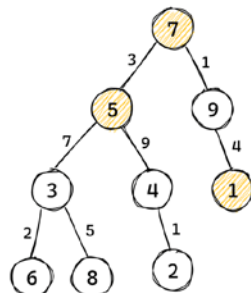
- 对于问题 $(1, 5)$ ，它们在1、2号树上的最近公共祖先分别是 $l_1 = 1$ ， $l_2 = 7$ ，因此交互库回答 $d_1(1, 1) = 0$ ， $d_1(1, 5) = 2$ ， $d_2(7, 1) = 5$ ， $d_2(7, 5) = 3$ 。
- 对于问题 $(1, 7)$ ，它们在1、2号树上的最近公共祖先分别是 $l_1 = 1$ ， $l_2 = 7$ ，因此交互库回答 $d_1(1, 1) = 0$ ， $d_1(1, 7) = 3$ ， $d_2(7, 1) = 5$ ， $d_2(7, 7) = 0$ 。

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

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



1



2



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) : $N = 1\,000\,000$, $Q = K - 1$



奖品（中欧信息学奥林匹克 CEOI 2022）

这个题刚刚读完的时候，感觉有一点莫名其妙，两棵树除了节点个数相同外，毫无关系，但我们每次询问和最后回答的问题，都基于两棵树上的同一批编号的节点。

难道是随便取 K 个点，随便问了 Q 个问题后，都能得出两两之间的所有距离关系？对于这个猜想，从直觉上似乎答案是否定的，否则这题就没必要搞两棵树这么复杂。

但仔细一想，这个猜想也似乎有点靠谱之处。注意到除了Subtask 2以外，都有 $Q = K - 1$ ，也就是问题数和节点树基本相同。设 $d(x)$ 表示 x 到根的距离，那么对于某一棵树而言，每次询问都能产生两个类似于 $d(x) - d(y) = w$ 的方程，共 $2K - 2$ 个方程（可能重复）；而一共有 $2K - 1$ 个未知数（ K 个点和 $K - 1$ 个LCA，也可能重复）。

请思考：这个猜想是否有反例？

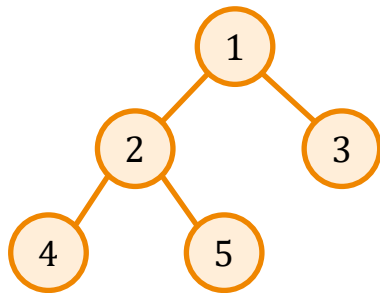


奖品（中欧信息学奥林匹克 CEOI 2022）

“随便取 K 个点，随便问了 Q 个问题后，都能得出两两之间的所有距离关系”的反例，如右图所示。

假设选取的 $K = 3$ 个点为 $\{3, 4, 5\}$ ，询问的 $Q = K - 1 = 2$ 个问题为 $(3, 4)$ 、 $(3, 5)$ ，那么我们只能得到节点 $3, 4, 5$ 到节点 1 的距离，对节点 2 到其它节点的距离毫不知情。

所以问题出在哪里呢？方程的个数是能达到未知数的个数的，所有的未知数也都解出来了。但我们最终需要的节点 2 ，却是整个方程组里缺失的。





奖品（中欧信息学奥林匹克 CEOI 2022）

那么，我们怎么“取 K 个点，问 Q 个问题”，才能保证集合中所有的节点和它们的LCA都在方程组中出现？这个题最难的点在于，我们有两棵树，取 K 个相同编号的节点，问 Q 个相同的问题，它们都要得到这样的保证。

于是我们考虑下面的两个问题：

- 是否存在一个 K 个节点的集合，（几乎）无论问哪 Q 个问题，都能保证所有节点和它们的LCA都在方程组中出现
- 是否存在一个提问的顺序/模式，无论选哪 K 个节点，都能保证所有节点和它们的LCA都在方程组中出现

如果以上两个问题都得到了解决，把它们合并起来，两棵树的问题就迎刃而解。

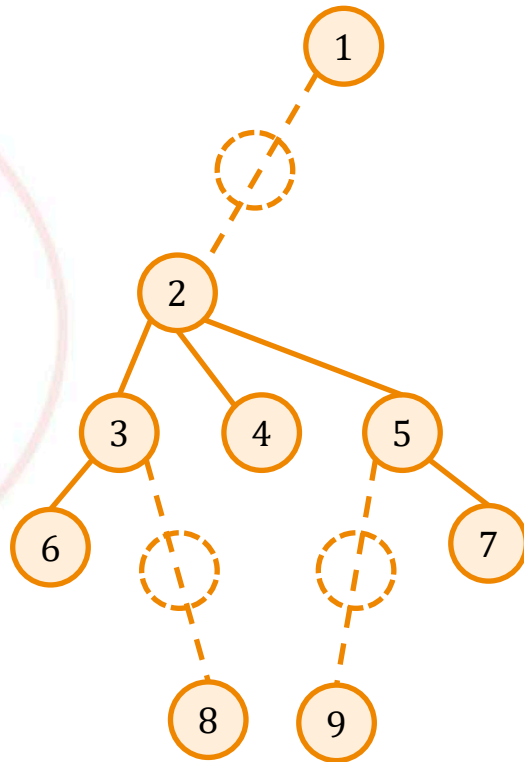


奖品（中欧信息学奥林匹克 CEOI 2022）

K个节点编号的选取：选第一棵树的前序遍历中的前K个节点的编号。

Q个问题的选取：设在第二棵树中，上述编号的节点按前序遍历的顺序排序，依次为 s_1, s_2, \dots, s_K ，那么我们向交互库提出的问题为 $(s_1, s_2), (s_2, s_3), \dots, (s_{K-1}, s_K)$ 。

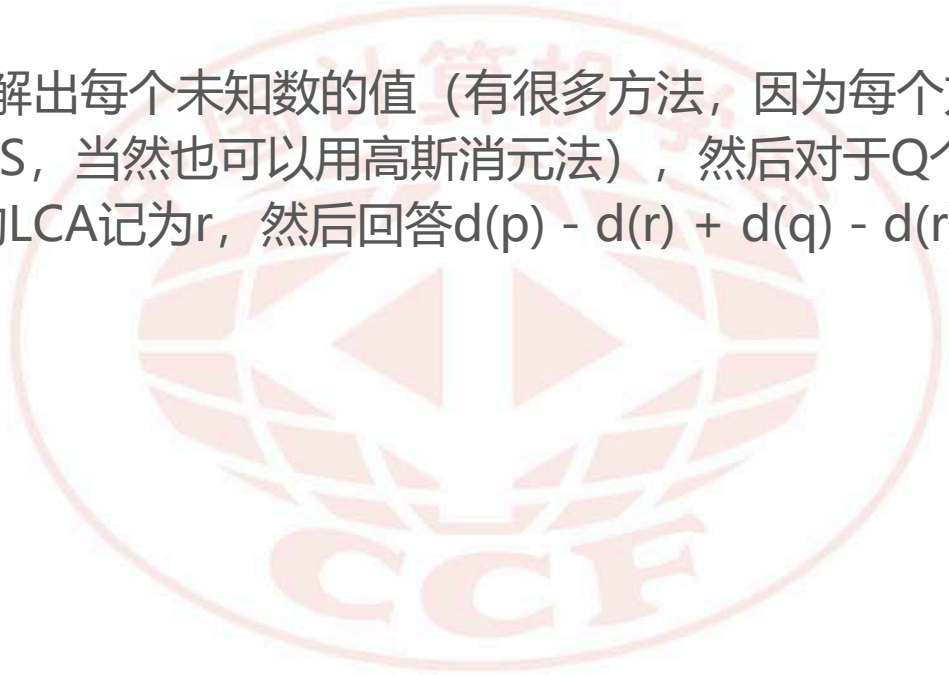
为什么这样的选取能保证所有的节点和它们的LCA一定在方程组中出现？比如节点8和节点9在K个节点的集合内，它们的LCA为节点2。只要我们是按前序遍历问的Q个问题，那么一定有一个问题里的两个节点位于节点2的两个不同子树，那么节点2作为它们的LCA，就出现在了方程组里。





奖品（中欧信息学奥林匹克 CEOI 2022）

最后，我们可以先解出每个未知数的值（有很多方法，因为每个方程只有2个未知数，所以可以用DFS，当然也可以用高斯消元法），然后对于Q个提问中的每一个(p, q)，找到它们的LCA记为r，然后回答 $d(p) - d(r) + d(q) - d(r)$ 即可。





中国计算机学会
China Computer Federation



奖品 (中欧信息学奥林匹克 CEOI 2022)

| ✓ | Rank | First Name | Last Name | Team | Abraça... | Homew... | Prize | CEOI 2022... | Drawing | Measures | Parking | CEOI 2022... | Global |
|---|------|----------------|----------------|------|-----------|----------|-------|--------------|---------|----------|---------|--------------|--------|
| | 1 | Dorijan | Lendvaj | | 100 | 100 | 100 | 300 | 40 | 100 | 100 | 240 | 540 |
| | 1 | Alexandru | Luchianov | | 100 | 100 | 100 | 300 | 40 | 100 | 100 | 240 | 540 |
| | 3 | Patrick | Pavić | | 100 | 100 | 76 | 276 | 25 | 59 | 100 | 184 | 460 |
| | 4 | Jan | Strzeszyński | | 10 | 100 | 76 | 186 | 40 | 100 | 100 | 240 | 426 |
| | 5 | Michał | Stawarz | | 100 | 100 | 19 | 219 | 0 | 100 | 100 | 200 | 419 |
| | 6 | Luca | Perju-Verzotti | | 100 | 100 | 19 | 219 | 0 | 59 | 100 | 159 | 378 |
| | 7 | Eliška | Macáková | | 100 | 100 | 10 | 210 | 10 | 100 | 20 | 130 | 340 |
| | 8 | Kacper | Paciorek | | 0 | 100 | 54 | 154 | 40 | 100 | 0 | 140 | 294 |
| | 9 | Andrei-Robert | Ion | | 100 | 100 | 10 | 210 | 0 | 45 | 20 | 65 | 275 |
| | 10 | István Ádám | Molnár | | 10 | 100 | 10 | 120 | 40 | 100 | 10 | 150 | 270 |
| | 11 | Alexandru-Raul | Todoran | | 100 | 100 | 10 | 210 | 0 | 24 | 20 | 44 | 254 |
| | 12 | Márton | Németh | | 35 | 100 | 0 | 135 | 15 | 0 | 100 | 115 | 250 |
| | 13 | Alisa | Potomkina | | 100 | 53 | 0 | 153 | 10 | 59 | 10 | 79 | 232 |
| | 14 | Benjamin | Bajd | | 10 | 100 | 10 | 120 | 0 | 100 | 10 | 110 | 230 |
| | 14 | Jakub | Konc | | 10 | 100 | 0 | 110 | 0 | 100 | 20 | 120 | 230 |
| | 14 | Lőrinc | Máté | | 10 | 100 | 10 | 120 | 10 | 0 | 100 | 110 | 230 |
| | 17 | Jeremiasz | Preiss | | 100 | 100 | 0 | 200 | 0 | 0 | 20 | 20 | 220 |
| | 18 | Marko | Dorčić | | 10 | 100 | 0 | 110 | 10 | 0 | 88 | 98 | 208 |
| | 19 | Lucas | Schwebler | | 10 | 100 | 0 | 110 | 0 | 59 | 28 | 87 | 197 |
| | 20 | Pál | Czanik | | 10 | 100 | 10 | 120 | 25 | 24 | 20 | 69 | 189 |
| | 21 | Massimiliano | Foschi | | 10 | 100 | 0 | 110 | 0 | 59 | 2 | 61 | 171 |
| | 22 | Štěpán | Mikéska | | 10 | 100 | 0 | 110 | 40 | 0 | 2 | 42 | 152 |
| | 23 | Daryna | Karpenko | | 10 | 100 | 10 | 120 | 0 | 0 | 20 | 20 | 140 |
| | 24 | Johann | Gaulke | | 10 | 100 | 0 | 110 | 25 | 0 | 2 | 27 | 137 |
| | 25 | Linus | Lüchinger | | 10 | 100 | 0 | 110 | 0 | 24 | 0 | 24 | 134 |



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：

| | | |
|----------|-------------------------|-----------------------|
| $M(q) =$ | 0 | $q > 250$ |
| | 0.7 | $250 \geq q \geq 225$ |
| | $0.7 + (225 - q) / 500$ | $225 > q > 125$ |
| | 1.0 | $125 \geq q$ |



最大公约数（西班牙信息学奥林匹克 OIE 2022）

首先我们观察 $\log_2 10^{18} \approx 60$ ，要得满分最多问125次，要得70%的分数最多问250次，意味着本题的解法大概和二分法有关。

我们先考虑问250次的方法。意味着，如果把 x 和 y 写成二进制，那么每个比特可以问4个问题。事实上，我们只要问如下4个问题，就能确定 x 和 y 的奇偶性：(0, 0), (0, 1), (1, 0), (1, 1)。

比如 x 是奇数， y 是偶数，那么只有当我们询问(1, 0)的时候，得到的最大公约数是偶数，询问其它3个问题，得到的都是奇数。于是，我们得到了 x 和 y 在二进制下的个位。

随后，我们询问 $(0 + a_0, 0 + b_0)$, $(0 + a_0, 2 + b_0)$, $(2 + a_0, 0 + b_0)$, $(2 + a_0, 2 + b_0)$ ，即可得知 x 和 y 二进制下的十位，其中 (a_0, b_0) 是刚刚算出来的 x 和 y 的个位。以此类推。



最大公约数（西班牙信息学奥林匹克 OIE 2022）

那么，如何只询问125次呢？这意味着 x 和 y 在二进制下的每一位只能询问2次。

在考察个位的4个问题 $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$ 中，不妨设只有在我们询问 $(1, 0)$ 的时候，得到的最大公约数是偶数，那么我们可以进一步考察它是不是4的倍数。

比如 $(x, y) = (5, 10)$ ，那么在考察个位询问 $(1, 0)$ 时，最大公约数为2，不是4的倍数，那么在考察十位询问 $(1, 0)$, $(1, 2)$, $(3, 0)$, $(3, 2)$ 时， $(1, 0)$ 和 $(3, 2)$ 事实上是不必询问的，因为询问的结果一定不可能是4的倍数。

再如 $(x, y) = (37, 24)$ ，在考察个位询问 $(1, 0)$ 时，最大公约数为12，是4的倍数，那么在考察十位询问 $(1, 0)$, $(1, 2)$, $(3, 0)$, $(3, 2)$ 时， $(1, 2)$ 和 $(3, 0)$ 事实上是不必询问的。

于是，除了第一位（个位）以外，我们每一位只需要询问2次即可。



最大公约数 (西班牙信息学奥林匹克 OIE 2022)

| Rank | Nombre | Apellidos | Team | pinta... | mcd | teatro | sumas | inspe... | Dia.1 | seque... | desor... | come... | vasos | reesc... | Dia.2 | Global |
|------|-------------|-------------------|------|----------|-------|--------|-------|----------|--------|----------|----------|---------|-------|----------|-------|--------|
| 1 | Dario | Martínez Ramírez | SEST | 30 | 91.76 | 100 | 30 | 24 | 275.76 | 85 | 100 | 93.12 | 0 | 100 | 378 | 653.76 |
| 2 | Hugo | Dominguez Santana | OIA | 30 | 3.64 | 100 | 100 | 60 | 293.64 | 100 | 100 | 21 | 0 | 100 | 321 | 614.64 |
| 3 | Manuel | Torres Cid | DI | 30 | 3.51 | 100 | 100 | 24 | 257.51 | 11 | 100 | 100 | 0 | 100 | 311 | 568.51 |
| 4 | Sergio | Domínguez Alonso | DI | 30 | 0 | 100 | 100 | 24 | 254 | 100 | 100 | 0 | 0 | 100 | 300 | 554 |
| 5 | Innokentiy | Kaurov | IM | 30 | 83.69 | 100 | 100 | 60 | 373.69 | 20 | 100 | 9 | 0 | 35 | 164 | 537.69 |
| 6 | Bernat | Pagès Vives | DI | 30 | 15 | 100 | 100 | 24 | 269 | 20 | 100 | 100 | 0 | 35 | 255 | 524 |
| 7 | Pau | Martí Biosca | IM | 30 | 0 | 100 | 100 | 43 | 273 | 20 | 100 | 26 | 0 | 100 | 246 | 519 |
| 8 | María Lucía | Aparicio García | | 30 | 5 | 100 | 100 | 41 | 276 | 20 | 100 | 0 | 0 | 100 | 220 | 496 |
| 9 | Oscar | Carballo Puebla | DI | 30 | 3.5 | 100 | 100 | 24 | 257.5 | 11 | 100 | 21 | 0 | 100 | 232 | 489.5 |
| 10 | Joan | Cintas Navarro | DI | 30 | 3.58 | 100 | 100 | 24 | 257.58 | 20 | 100 | 7.83 | 0 | 100 | 228 | 485.58 |
| 11 | Alejandro | Vivero Puga | DI | 30 | 0 | 100 | 100 | 24 | 254 | 20 | 100 | 0 | 0 | 100 | 220 | 474 |
| 12 | Daniel | Lopez Piris | DI | 30 | 0 | 100 | 55 | 43 | 228 | 20 | 100 | 8 | 0 | 100 | 228 | 456 |
| 13 | Pablo | Sáez Reyes | DI | 30 | 3.51 | 100 | 100 | 41 | 274.51 | 20 | 100 | 9 | 0 | 35 | 164 | 438.51 |
| 14 | Huize | Mao | DI | 0 | 3.51 | 100 | 100 | 13 | 216.51 | 0 | 100 | 9 | 0 | 100 | 209 | 425.51 |
| 15 | Aitor | Pfluegl Hernaez | DI | 0 | 3.51 | 100 | 100 | 24 | 227.51 | 11 | 36 | 0 | 0 | 100 | 147 | 374.51 |
| 16 | Luis | Gutiérrez Garrido | DI | 30 | 0 | 100 | 15 | 24 | 169 | 20 | 100 | 9 | 0 | 66 | 195 | 364 |
| 17 | Julio | Meroño Sáez | DI | 0 | 3.51 | 100 | 25 | 24 | 152.51 | 20 | 73 | 6.3 | 0 | 100 | 199 | 351.51 |
| 18 | Adriana | Aguiló Martínez | DI | 30 | 0 | 100 | 15 | 43 | 188 | 3 | 36 | 0 | 0 | 100 | 139 | 327 |
| 19 | Hugo | Lladró Prats | DI | 30 | 45 | 100 | 0 | 24 | 199 | 0 | 100 | 0 | 0 | 0 | 100 | 299 |
| 20 | Rafael | Torres Arrechea | OIA | 30 | 0 | 100 | 30 | 24 | 184 | 0 | 100 | 9 | 0 | 0 | 109 | 293 |
| 21 | Roger | Lidón Ardanuy | DI | 30 | 0 | 100 | 15 | 24 | 169 | 20 | 100 | 0 | 0 | 0 | 120 | 289 |
| 22 | Alberto | Navalón Lillo | SEST | 0 | 0 | 100 | 40 | 41 | 181 | 11 | 36 | 0 | 0 | 0 | 47 | 228 |
| 23 | Marcos | Domínguez Gómez | DI | 30 | 0 | 100 | 0 | 24 | 154 | 0 | 36 | 0 | 0 | 35 | 71 | 225 |
| 24 | Joan | Gomà Cortés | DI | 0 | 0 | 100 | 0 | 24 | 124 | 11 | 73 | 0 | 0 | 0 | 84 | 208 |

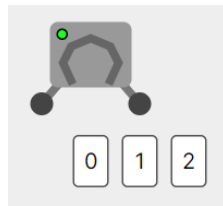
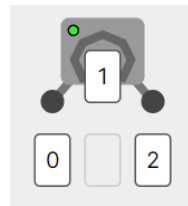
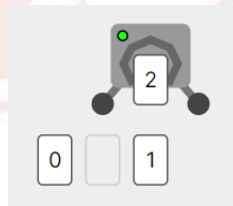
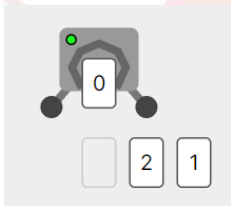
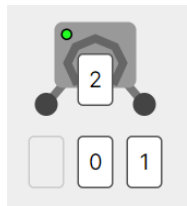
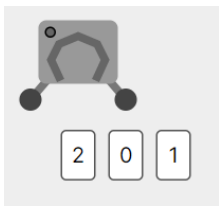


爪式排序 (瑞士信息学奥林匹克 SOI 2021/2022)

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):

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

| 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 |



爪式排序 (瑞士信息学奥林匹克 SOI 2021/2022)

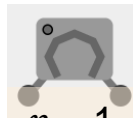
Subtask 3: 右侧是一种排序方法。

我们每次把当前最大、次大的两个元素挪到最右边，把最小、次小的挪到最左边，然后剩下 $n - 4$ 个元素，再递归处理。

只要按下述模式“平移”操作，就能在爪子带着最大元素的情况下，把次大元素挪到最右边。最小、次小元素同理。



$[n - 1, n - 2, \dots, 1, 0]$



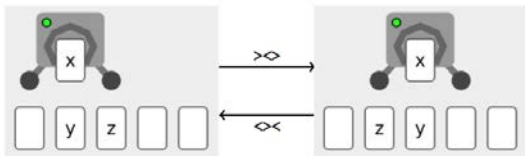
$[\text{空}, n - 2, \dots, 1, 0]$



$[\text{空}, n - 3, n - 4, \dots, 2, 1, n - 2, n - 1]$



$[0, 1, n - 3, n - 4, \dots, 2, n - 2, n - 1]$





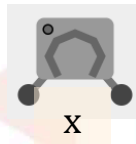
爪式排序 (瑞士信息学奥林匹克 SOI 2021/2022)

Subtask 4: 我们参考Subtask 3的解法, 先把最大的两个元素挪到最右侧。

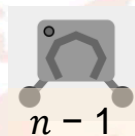
先考虑最大的元素在次大元素左侧的情况, 我们先把爪子移到最大元素上方, 然后把它“平移”到次大元素相邻的位置, 即可按Subtask 3的方式把它们挪到最左侧的位置。其余情况类似处理。



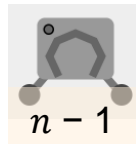
空
 $[..., n-1, ..., n-2, ...]$



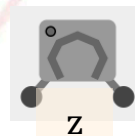
x
 $[..., n-1, n-2, ...]$



$n-1$
 $[...y, z, n-2]$



$n-1$
 $[..., x, n-2, ...]$



z
 $[...y, n-2, n-1]$



爪式排序 (瑞士信息学奥林匹克 SOI 2021/2022)

| Rank | Username | Total (500) | tshirts (100) | clawsort (100) | dancing (100) | rerouting (100) | tinmining (100) |
|------|-----------------------|----------------|------------------|-------------------|------------------|--------------------|--------------------|
| 1 | Elias Bauer | 482 | 100 | 97 | 100 | 100 | 85 |
| 2 | Vivienne Burckhardt | 480 | 100 | 95 | 100 | 100 | 85 |
| 3 | Josia John | 449 | 100 | 90 | 100 | 92 | 67 |
| 4 | Nandor Kovacs | 404 | 100 | 82 | 100 | 92 | 30 |
| 5 | Jasmin Studer | 400 | 100 | 82 | 43 | 100 | 75 |
| 6 | Pascal Gamma | 345 | 100 | 82 | 43 | 60 | 60 |
| 7 | Linus Lüchinger | 330 | 100 | 99 | 31 | 100 | |
| 8 | Lukas Münzel | 325 | 100 | 92 | 33 | 100 | |
| 9 | Hannah Oss | 303 | 100 | 93 | 100 | 10 | |
| 10 | Yaël Arn | 300 | 100 | 100 | 100 | | |
| 11 | Jérémie Pierre | 296 | 100 | 93 | 43 | 30 | 30 |
| 12 | Ursus Wigger | 293 | 100 | 93 | 100 | | |
| 12 | Priska Steinebrunner | 293 | 100 | 90 | 43 | 60 | |
| 14 | Noah Bodmer | 289 | 100 | 86 | 43 | 60 | 0 |
| 15 | Till Bégue | 250 | 100 | 75 | 15 | 60 | |
| 16 | Andrei-Octavian Dirla | 230 | 60 | 70 | 100 | | |
| 17 | Jan Wilhelm | 229 | 100 | 86 | 43 | | |
| 17 | Lucien Kissling | 229 | 100 | 86 | 43 | | |
| 19 | Ferdinand Ornskov | 221 | 100 | 78 | 43 | | |
| 20 | Robin Steiner | 218 | 100 | 75 | 43 | | |
| 21 | Lionel Müller | 205 | 100 | 90 | 15 | | |
| 22 | Ian Wasser | 203 | 100 | 88 | 15 | | |
| 22 | Florian Donnelly | 203 | 100 | 98 | 5 | 0 | |



中国计算机学会
China Computer Federation



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：)