

D. Laser Strike

题目名称	激光突击
时间限制	3 秒
空间限制	1 GB

Ann 和她的朋友 Kathrin 最近迷上了一款新桌游：《激光突击》。她们俩非常喜欢这款游戏，甚至成了她们的最爱。在这款游戏中，两名玩家需要合作，将棋盘上的 N 个棋子全部移除。游戏分为两个阶段。但关键在于：Kathrin 并不能看到全部的信息。为了赢得游戏，Ann 和 Kathrin 必须尽可能少地交流信息，同时进行有效配合。

棋盘上有 N 个编号为 0 到 $N - 1$ 的互不相同的棋子，两位玩家都可以看到这些棋子。此外，还存在 $N - 1$ 条连接不同棋子的边，使得任意两个棋子之间都可以通过这些连接到达。换句话说，这些连接构成了一棵树。但只有 Ann 能看到这些连接，Kathrin 看不到它们。

在游戏的第一阶段，Ann 决定一组移除棋子的顺序 $\ell_0, \ell_1, \dots, \ell_{N-2}$ ，直到只剩下一个棋子。这个顺序对 Kathrin 是保密的，如果她能准确复现这个顺序，两人就可以赢得游戏。移除棋子时必须满足以下规则：每次移除的棋子，必须与当前仍在棋盘上的某一个棋子直接相连，且这两个棋子之间只有这一条连接。换句话说，每次被移除的棋子，必须是当前剩余图中的叶子结点。（当 $N - 1$ 个棋子被移除后，最后一个棋子会自动移除，游戏结束。）Ann 必须选择一个满足上述规则的合法移除顺序。

随后，Ann 会写下一条消息传达给 Kathrin，这条消息是一个二进制字符串。Ann 可以自由决定消息的长度——但消息越短，她们得到的分数就越高。

接下来进入游戏的第二阶段。Kathrin 的目标是依次移除 $N - 1$ 个棋子，顺序必须为 $\ell_0, \ell_1, \dots, \ell_{N-2}$ 。她将进行 $N - 1$ 次操作。

在第 i 次操作之前，Ann 会告诉 Kathrin 一对整数 a, b ，满足以下条件：

- $a < b$;
- 当前棋盘上仍有编号为 a 和 b 的两个棋子，且它们直接连接；
- 在这一轮操作中， a 或 b 中有一个是应当被移除的棋子 ℓ_i 。

注意：对于 Ann 来说，在当前的树中，某一片叶子 ℓ_i 所连接的边 (a, b) 是唯一确定的。

然后，Kathrin 会从 a 和 b 中选择一个棋子将其移除。如果她选择的是正确的棋子（即 ℓ_i ），则游戏继续进行；否则游戏立即失败。

你的任务是设计并实现 Ann 和 Kathrin 的策略，使得她们能够总是赢得游戏。

本题的评分将依据 Ann 在第一阶段所写的消息长度：消息越短，得分越高。

实现说明

这是一个多轮运行的交互题，你的程序将被执行两次。第一次执行时，你的程序应实现 Ann 的策略（游戏的第一阶段）；第二次执行时，你的程序应实现 Kathrin 的策略（游戏的第二阶段）。

输入的第一行包含两个整数 P 和 N ： P 为 1 或 2，表示当前是第一阶段（Ann）还是第二阶段（Kathrin）； N 是棋子的数量。

接下来的输入格式将根据 P 的值有所不同：

第一阶段: Ann

继首行（见前述）之后的接下来的 $N - 1$ 行描述了树的结构。每行包含两个整数 a 和 b ，表示棋子 a 和 b 之间存在一条连接（ $0 \leq a < b \leq N - 1$ ）。

你的程序应首先输出一行 Ann 要写给 Kathrin 的信息，形式为一个由不超过 1000 个 0 或 1 组成的二进制字符串。如果你想输出一个长度为 0 的字符串，可以直接输出一个空行。

随后，你的程序应输出 $N - 1$ 个整数 $\ell_0, \ell_1, \dots, \ell_{N-2}$ ，每个占一行，表示 Ann 希望 Kathrin 依次移除的棋子编号。这个移除顺序必须满足：在每一步中，被移除的棋子必须是当前树中的一个叶子节点，也就是说，每次移除后，剩下的棋子构成的图仍然是一棵树。

第二阶段: Kathrin

在第一行输入（见前述）之后，下一行输入包含一个二进制字符串，即第一阶段 Ann 所写的信息。

接下来将进行 $N - 1$ 轮交互，每轮对应 Kathrin 的一次操作。

在第 i 轮中，你的程序应首先读入两个整数 a 和 b （ $0 \leq a < b \leq N - 1$ ）。这两个棋子中，一个是当前阶段应被移除的叶子 ℓ_i ，另一个是当前仍与该叶子直接相连的唯一一个棋子。

然后，你的程序应输出一个整数 ℓ_i ，表示 Kathrin 选择移除的棋子编号。如果输出的不是正确的 ℓ_i ，则游戏失败，你的提交将被判为 Wrong Answer。

细节说明

如果你的程序在两次运行中的总运行时间超过时间限制，你的提交将被判为 Time Limit Exceeded（超时）。

请务必在每次输出后刷新标准输出缓冲区，否则可能会被判为超时。

在 Python 中，只要使用 `input()` 读取输入，系统会自动刷新输出。

在 C++ 中，使用 `cout << endl`；可以在输出换行的同时刷新缓冲区；若使用 `printf`，请调用 `fflush(stdout)`；手动刷新。

约束条件和评分

- $N = 1\,000$ 。
- 对于所有连接 $0 \leq a < b \leq N - 1$ 。

你的解法将会在若干个数据组上进行评测，每个数据组对应一定的分值。每个数据组包含若干个测试点，只有在该组的所有测试点均通过时，才能获得该组对应的分数。

数据组	最高分数	额外约束条件
1	8	树是一个星形结构。也就是说，除了一个节点以外，其余所有节点都是叶子节点。
2	9	树是一个链状结构。也就是说，除了两个叶子节点外，其余所有节点都恰好有两个相邻节点
3	21	树是一个星状结构，并从中心向外延伸出链条。也就是说，除了一个拥有多个相邻节点的节点外，其他所有节点都只有一个或两个相邻节点。
4	36	任意两个节点之间的距离最多为 10。
5	26	无特殊约束

对于每个通过的数据组，你将根据以下公式获得分数：

$$\text{score} = S_g \cdot (1 - 0.3 \cdot \log_{10} \max(K, 1)),$$

其中， S_g 是该数据组的最高分数， K 是在该数据组中 Ann 在任意测试点的消息所用的最大长度。你在每个测试组中的得分将四舍五入到最接近的整数。

下表显示了当你的程序在通过所有测试组时，且最大 K 为特定值时你能获得的总分数。如果要获得满分 100，你的程序必须能在所有测试用例中都保证 $K \leq 1$ 。

K	1	5	10	50	100	500	1000
Score	100	79	70	49	39	20	11

测试工具

为了便于你调试程序，我们提供了一个简单的测试工具，你可以在 Kattis 题目页面底部的“attachments”部分下载。该工具为可选使用，注意：正式的 Kattis 评测器与该测试工具实现方式可能不同。

使用方法如下：创建一个输入文件，例如 sample1.in。该文件应以一个整数 N 开头，接下来 $N - 1$ 行描述树的结构，格式与第一阶段（Phase 1）中的格式相同。

例如，对于下面的示例：

```
7
0 1
1 2
2 3
0 4
0 6
1 5
```

对于 Python 程序（如 solution.py，通常使用 pypy3 solution.py 运行），可以使用以下命令运行测试工具：

```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

对于 C++ 程序，首先编译（例如使用以下命令）：

```
g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out
```

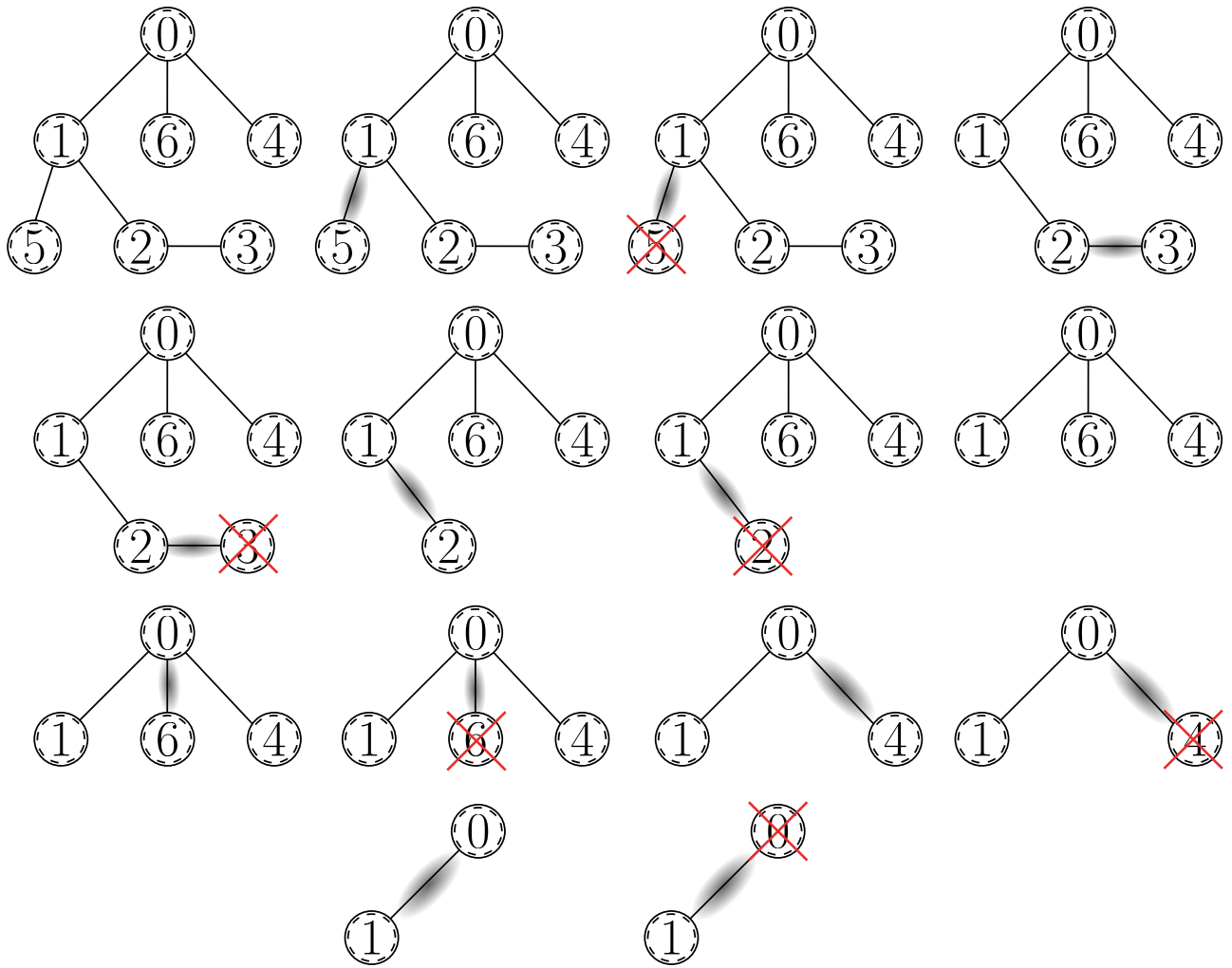
然后运行测试工具：

```
python3 testing_tool.py ./solution.out < sample1.in
```

样例

请注意，本节中的样例使用 $N = 7$ 是为了简化说明，因此并不是一个合法的测试点。你的程序不需要处理此样例。在实际测评中，所有测试点都将满足 $N = 1000$ 。

在这个样例中，Ann 拿到如图下这棵树。在第一阶段，Ann 读入这棵树后，选择了一个二进制字符串 "0110" 发送给 Kathrin，并选定了一个顶点移除顺序 $[\ell_0, \ell_1, \dots, \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$ ，也就是说她期望 Kathrin 按照这个顺序从树中移除点。在第二阶段，Kathrin 收到了第一阶段发送来的字符串 "0110"，之后每次会接收到一对叶子结点，例如第一对是 (1, 5)，她决定移除叶子结点 5。接着她收到下一对 (2, 3)，她选择移除结点 3，如此反复。下图展示了整个交互过程的各个阶段：



评测器输出	你的输出
1 7	
0 1	
1 2	
2 3	
0 4	
0 6	
1 5	
	0110
	5
	3
	2
	6
	4
	0

评测器输出	你的输出
2 7	
0110	
1 5	
	5
2 3	
	3
1 2	
	2
0 6	
	6
0 4	
	4
0 1	
	0