

D. レーザー・ストライク (Laser Strike)

問題名	レーザー・ストライク (Laser Strike)
実行時間制限	3 秒
メモリ制限	1 GB

Ann と友達の Kathrin は、最近新しいボードゲーム「Laser Strike」に興味を持った。このゲームでは、 N 個の駒を盤面から除去することが目標であり、そのために 2 人のプレイヤーが協力する。このゲームは 2 つのフェーズからなる。ここで重要なことは、Kathrin はゲームに関する完全な情報を知らないことである。ゲームで勝利するには、Ann と Kathrin は互いに協力する必要がある一方、互いのコミュニケーションの量をできるだけ少なくしなければならない。

盤面には N 個の異なる駒があり、それぞれ 0 から $N - 1$ までの番号が付けられている。駒は両者が見ることができる。また、駒の間には $N - 1$ 個の辺があり、どの駒からもどの駒へも、何本かの辺をたどって到達することができる。すなわち、辺は木構造をなす。ここで、**Ann のみが辺の情報を知っており、Kathrin は知らない。**

ゲームのフェーズ I では、Ann が (残りの駒が 1 つになるまでの) 除去の順番 $\ell_0, \ell_1, \dots, \ell_{N-2}$ を決める。この順番は Kathrin には知らされず、もし Kathrin がこの順番を正しく復元することができたら、ゲームに勝利する。ここで、駒を除去する順番は以下のルールを満たす必要がある:

- 駒が除去される時、除去される駒は、残りの駒のうちちょうど 1 個と辺で繋がれていなければならない。言い換えると、除去される駒は (除去される駒と残りの駒から構成される木の) 葉でなければならない (ここで、 $N - 1$ 個の駒が除去されると、最後の駒は自動的に除去され、プレイヤーは勝利する)。

Ann は上記のルールに従った順番を選択しなければならない。

加えて、Ann は Kathrin に向けて 0 と 1 からなる文字列で表されるメッセージを書き込む。Ann は文字列の長さを自由に選択することができるが、短ければ短いほど、高得点が得られる。

その後、フェーズ II が始まる。このゲームの目標は、Kathrin が $\ell_0, \ell_1, \dots, \ell_{N-2}$ の順番で $N - 1$ 個の駒を除去することである。Kathrin は $N - 1$ 手を指す。 i 手目の直前には、Ann は以下の条件を満たす 2 つの整数 a, b を Kathrin に伝える:

- $a < b$.
- 駒 a と駒 b は辺で直接結ばれており、両方ともまだ残っている。
- 駒 a と駒 b のうち片方は、今回取り除かれるべき駒 ℓ_i である。

なお、Ann が伝える (a, b) は ℓ_i から一意に定まることが証明できる。

その後、Kathrin は駒 a と駒 b のうちいずれか片方を除去する。もしこれが正しい駒 ℓ_i であれば、ゲームが続行し、そうでなければ敗北となる。

あなたの課題は、Ann と Kathrin がゲームで勝利するための戦略を実装することである。

あなたのプログラムは、フェーズ I で Ann が残したメッセージの長さによって採点される。

実装の詳細

この問題は複数回実行の問題である。すなわち、あなたのプログラムは複数回実行される。最初の実行では、フェーズ I における Ann の戦略を実装しなければならない。2 回目の実行では、フェーズ II における Kathrin の戦略を実装しなければならない。

入力の最初の行は 2 つの整数 P, N からなる。ここで P は 1 または 2 であり (フェーズ I, II のどちらであるかを表す)、 N は駒の数である。

その後の入力は、現在のフェーズに依存する。

フェーズ I: Ann

最初の行 (前述) を受け取った後、木の構造を表す $N - 1$ 行を受け取る。各行は 2 つの整数 a, b ($0 \leq a < b \leq N - 1$) からなり、駒 a と駒 b の間に辺があることを意味する。

あなたのプログラムは最初に、(Ann が書くメッセージに対応する) 0 と 1 からなる長さ 1 000 以下の文字列を出力しなければならない。ここで、長さ 0 の文字列を出力したい場合は、空行 (何もない行) を出力しなければならない。

その後、あなたは $N - 1$ 個の整数 $\ell_0, \ell_1, \dots, \ell_{N-2}$ を各行 1 個ずつ出力しなければならない。これらの整数は、Ann が駒を除去する順番に対応する。ここで、駒が指定された順番で 1 つずつ除去していったとき、除去する駒は木の葉でなければならない。すなわち、木は常に連結でなければならない。

フェーズ II: Kathrin

最初の行 (前述) を受け取った後の次の行は、0 と 1 からなる文字列 (Ann がフェーズ I で残したメッセージ) である。

その後、 $N - 1$ 回のインタラクションを行う。各回は、Kathrin の打つ手に対応する。

i 手目では、あなたのプログラムはまず 2 つの整数 a, b ($0 \leq a < b \leq N - 1$) を受け取る。駒 a, b のうち片方は (Ann の順番における) 葉 ℓ_i であり、もう片方は残っている駒の中で ℓ_i と繋がる唯一の駒である。その後、あなたのプログラムは正解 ℓ_i を出力しなければならない。もしプログラムが正しい葉 ℓ_i を出力しなかった場合、ゲームは敗北となり、あなたのプログラムは Wrong Answer と判定される。

詳細

もし 2 回の実行における実行時間の 総和 が実行時間制限を超過した場合、Time Limit Exceeded と判定される。

各行の出力の最後には、必ず標準出力を flush せよ。さもなくば、Time Limit Exceeded と判定される。Python の場合、`input()` によって自動的に flush される。C++ の場合、`cout << endl;` によって、改行されるとともに自動的に flush される。もし `printf` を使用する場合、`fflush(stdout);` を用いよ。

ここで、空行を正しく入力する処理はトリッキーな場合があるが、今回提供されるテンプレートでは、正しく処理することができる。

制約・得点

- $N = 1000$.
- すべての辺に対し $0 \leq a < b \leq N - 1$.

あなたの解答は各小課題ごとに評価され、小課題にはそれぞれ配点が割り当てられている。各小課題は複数のテストケースからなる。各小課題について得点を得るためには、その小課題に含まれるすべてのテストケースに正解する必要がある。

小課題	配点	制約
1	8	木はスター型をなす。厳密には、1 個の駒を除き、すべて葉である。
2	9	木はライン型をなす。厳密には、2 個の駒は葉であり、それ以外の駒は次数が 2 である。
3	21	木はスター型であるが、中心から出る線の長さは 1 とは限らない。厳密には、1 つの駒だけ次数 3 以上を持つ一方、それ以外の駒は次数 1 または 2 を持つ。
4	36	どの 2 つの駒の間も、距離が 10 以下である。
5	26	追加の制約はない。

正解した各小課題について、次式に基づく得点が与えられる。

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

ここで、 S_g は各小課題の配点であり、 K はこの小課題を満たすケースで Ann が残したメッセージの最大の長さである。**各小課題におけるあなたの得点は、最も近い整数に四捨五入される。**

以下の表には、いくつかの K の値について、もしあなたのプログラムがすべての小課題でこのような K の値となった場合の得点が示されている。特に、この問題で 100 点を得るためには、すべてのテストケースについて $K \leq 1$ で解く必要がある。

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

テストのためのツール

あなたが解法をテストすることを容易にするため、簡単なツールがダウンロードできるようになっている。Kattis 問題ページの下部の "attachments" の項を見よ。ここで、本ツールを必ずしも使う必要はない。なお、Kattis で使用される実際の採点プログラムは、本ツールとは異なる。

本ツールを使うには、まず "sample1.in" のような入力ファイルを作成しなければならない。この入力ファイルでは、最初の行に整数 N ，次の $N - 1$ 行に木の情報が、フェーズ I と同じ形式で入っていないといけない。具体例を以下に示す。

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

Python の場合、例えば `solution.py` の場合、(通常は `python3 solution.py` とすれば実行されるが) 以下を実行しなければならない。

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

C++ の場合、まずコンパイルを行い (例: `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`)，その後以下を実行しなければならない。

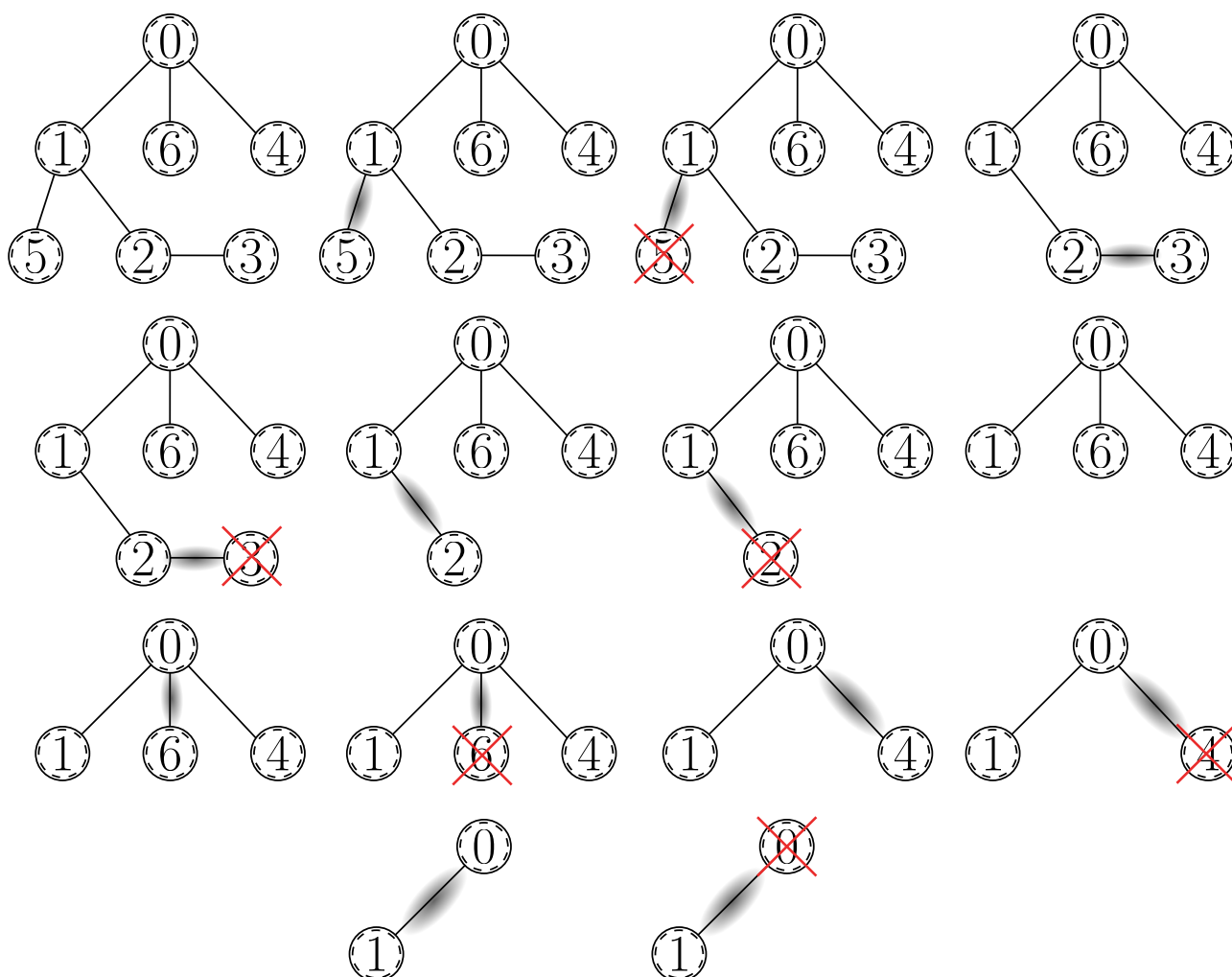
```
python3 testing_tool.py ./solution.out < sample1.in
```

例

まず、説明のため、例は $N = 7$ となっているが、これは制約を満たすテストケースではない。あなたのプログラムは必ずしもこの入力に正解する必要はなく、採点に使われるすべてのテストケースは $N = 1000$ を満たす。

この例では、Ann には下図のような木が入力される。フェーズ I では、Ann は木を読み込み、メッセージ "0110" を Kathrin に送信し、除去の順番 $[\ell_0, \ell_1, \dots, \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$ を指定する。フェーズ II では、Kathrin はまずフェーズ I で送信された文字列 "0110" を入力する。その後、彼女は整数 (1, 5) を

受け取り，駒5を除去すると決める（これは実際に葉である）．次の手では，彼女は整数 $(2, 3)$ を受け取り，葉である駒3を除去する．以降も同様に続く．下図はインタラクションの様子を示している．



採点プログラムの出力	あなたのプログラムの出力
1 7	
0 1	
1 2	
2 3	
0 4	
0 6	
1 5	
	0110
	5
	3
	2
	6
	4
	0

採点プログラムの出力	あなたのプログラムの出力
2 7	
0110	
1 5	
	5
2 3	
	3
1 2	
	2
0 6	
	6
0 4	
	4
0 1	
	0