

## D. Laser Strike

| Nombre del problema | Laser Strike |
|---------------------|--------------|
| Límite de tiempo    | 3 segundos   |
| Límite de memoria   | 1 gigabyte   |

Ann y su amiga Kathrin han descubierto recientemente un nuevo juego de mesa que se ha convertido en su favorito: Laser Strike. En este juego, los dos jugadores cooperan para quitar  $N$  piezas de un tablero. El juego funciona en dos fases. La gracia es que Kathrin no tendrá información completa sobre el juego. Para ganar el juego, Ann y Kathrin trabajarán juntas, mientras que se comunican lo mínimo posible.

Hay  $N$  piezas únicas en el tablero, numeradas de 0 a  $N - 1$ . Ambos jugadores pueden ver estas piezas. También hay  $N - 1$  conexiones entre pares de piezas, de forma que es posible alcanzar cualquier pieza desde otra siguiendo estas conexiones. En otras palabras, las conexiones forman un árbol. **Solo Ann puede ver las conexiones; Kathrin no las conoce.**

En la primera fase del juego, Ann decide un orden  $\ell_0, \ell_1, \dots, \ell_{N-2}$  en el cual las primeras  $N - 1$  piezas deberían de ser eliminadas. El orden será desconocido para Kathrin. Si lo puede replicar, ambas ganarán el juego. La eliminación de las piezas debe satisfacer la siguiente regla: cada vez que se elimina una pieza, debe estar conectada a exactamente otra pieza. En otras palabras, debe ser la hoja de un árbol formada por las piezas restantes. (Después de que se eliminen  $N - 1$  piezas, la última pieza se elimina automáticamente y los jugadores ganan.) Ann debe elegir un orden que respete la regla anterior.

Ann también escribirá un mensaje a Kathrin, en la forma de una cadena binaria. Ann puede elegir la longitud del mensaje – cuanto más corto sea, más puntos obtendrán.

Después de esto, la segunda fase del juego comienza. El objetivo es que Kathrin elimine  $N - 1$  piezas del tablero en el orden  $\ell_0, \ell_1, \dots, \ell_{N-2}$ . Realizará  $N - 1$  movimientos. Antes del movimiento  $i$ , Ann le dirá a Kathrin una pareja de enteros  $a, b$  con las siguientes propiedades:

- $a < b$ ;
- existe un par de piezas conectadas  $a$  y  $b$ ; y
- o bien  $a$  o bien  $b$  son la pieza correcta  $\ell_i$  que debería ser eliminada en este movimiento.

Para Ann la conexión  $(a, b)$  está determinada únicamente por la hoja  $\ell_i$  en el árbol actual.

Kathrin elimina o bien  $a$  o bien  $b$  del tablero. Si es la pieza correcta – es decir,  $\ell_i$  – continuarán jugando. De lo contrario pierden el juego.

Tu deber será implementar las estrategias de Ann's y Kathrin's para ganar el juego.

Tu programa será puntuado dependiendo de la longitud del mensaje que Ann escriba en la primera fase del juego.

## Implementación

Este es un problema de múltiples ejecuciones, lo que significa que tu programa será llamado dos veces. La primera vez que es ejecutado, debería implementar la estrategia de Ann para la primera fase del juego. Después, debería implementar la estrategia de Kathrin para la segunda fase del juego.

La primera línea contiene dos enteros,  $P$  y  $N$ , donde  $P$  es o bien 1 o bien 2 (primera o segunda fase), y  $N$  el número de piezas.

La entrada siguiente dependerá de la fase:

### Fase 1: Ann

Tras la primera línea (descrita anteriormente) las siguientes  $N - 1$  líneas de la entrada describen un árbol. Cada línea contiene dos enteros,  $a$  y  $b$  ( $0 \leq a < b \leq N - 1$ ), indicando una conexión entre las piezas  $a$  y  $b$ .

Tu programa deberá imprimir una cadena binaria con 1 000 caracteres como máximo, cada 0 o 1, del mensaje escrito por Ann.

Después de esto, debería imprimir  $N - 1$  enteros  $\ell_0, \ell_2, \dots, \ell_{N-2}$  en líneas separadas, indicando el orden en el cual Ann quiere eliminar hojas del árbol. El orden debe ser tal que si las piezas son eliminadas una a una del árbol, la pieza eliminada deberá ser una hoja, es decir, el árbol debe mantenerse conectado en todo momento.

### Fase 2: Kathrin

Tras la primera línea (descrita anteriormente), la siguiente línea contiene la cadena binaria (el mensaje de Ann) de la fase 1.

Después de esto, habrá  $N - 1$  rondas de interacción, una para cada movimiento de Kathrin.

En el movimiento  $i$ -ésimo, tu programa deberá leer dos enteros,  $a$  y  $b$  ( $0 \leq a < b \leq N - 1$ ). Una de estas piezas es  $\ell_i$  en el orden determinado por Ann, y la otra pieza es la única pieza conectada a  $\ell_i$ . Entonces, tu programa debería imprimir  $\ell_i$ , indicando que Kathrin elimina esta hoja. Si este

programa no imprime la hoja correcta  $\ell_i$ , las chicas pierden el juego y tu envío obtendrá el veredicto Wrong Answer.

## Detalles

Si la *suma* de los tiempos de ejecución de las dos ejecuciones de tu programa excede el tiempo límite, tu envío recibirá el veredicto de Time Limit Exceeded.

Asegurate de hacer flush por la salida estándar tras imprimir cada línea, o tu programa podrá juzgarse como Time Limit Exceeded. En Python, esto ocurre automáticamente siempre y cuando utilices `input()` para leer las líneas. En C++, `cout << endl;` hace flush a la vez que imprime una nueva línea; en el caso de utilizar `printf`, utiliza `fflush(stdout);`.

## Restricciones y Puntuación

- $N = 1\,000$ .
- $0 \leq a < b \leq N - 1$  para todas las conexiones.

Tu solución será verificada en un grupo de casos de prueba, cada uno con una puntuación correspondiente. Cada grupo contiene un conjunto de casos de prueba. Para obtener los puntos del grupo, debes resolver todos los casos del grupo.

| Grupo | Puntuación máxima | Restricciones   |
|-------|-------------------|---|
| 1     | 8                 | El árbol es una estrella. Es decir, todos los nodos excepto 1 son hojas.  |
| 2     | 9                 | El árbol es una línea. Es decir, todos los nodos excepto dos tienen exactamente dos nodos adyacentes.   |
| 3     | 21                | El árbol es una estrella con líneas saliendo de él. Es decir, todos los nodos tienen uno o dos nodos adyacentes, excepto uno que tiene más de dos nodos adyacentes. |
| 4     | 36                | La distancia entre dos nodos es como máximo 10.   |
| 5     | 26                | Sin restricciones adicionales.  |

Por cada grupo que tu programa resuelve correctamente, vas a recibir una puntuación basada en la siguiente fórmula:

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

donde  $S_g$  es la puntuación máxima del grupo, y  $K$  es la longitud máxima del mensaje mandado por Ann por cada caso de prueba en el grupo (acotada inferiormente por 1). Tu puntuación para cada test va a ser redondeada al entero más cercano.

La tabla de abajo muestra el número de puntos, para algunos valores de  $K$ , que tu programa va obtener si resuelve todos los grupos de prueba con ese valor de  $K$ . En particular, para conseguir una puntuación de 100 puntos, tu solución debe resolver cada caso de prueba con  $K \leq 1$ .

| K          | 1   | 5  | 10 | 50 | 100 | 500 | 1000 |
|------------|-----|----|----|----|-----|-----|------|
| Puntuación | 100 | 79 | 70 | 49 | 39  | 20  | 11   |

## Herramienta de Testing

Para facilitar la verificación de tu solución, tenemos una herramienta simple que puedes descargar. Mira “attachments” al fondo de la página del problema de Kattis. El uso de la herramienta es opcional. El grader oficial de Kattis difiere de la herramienta.

Para utilizar la herramienta, crea un fichero de entrada, como “sample1.in”, que debería comenzar con un entero  $N$  seguido de  $N - 1$  líneas describiendo el árbol, en el mismo formato que la fase 1. Para el siguiente ejemplo:

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

Para programas de Python, por ejemplo `solution.py` (normalmente ejecutados como `pypy3 solution.py`), ejecuta:

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

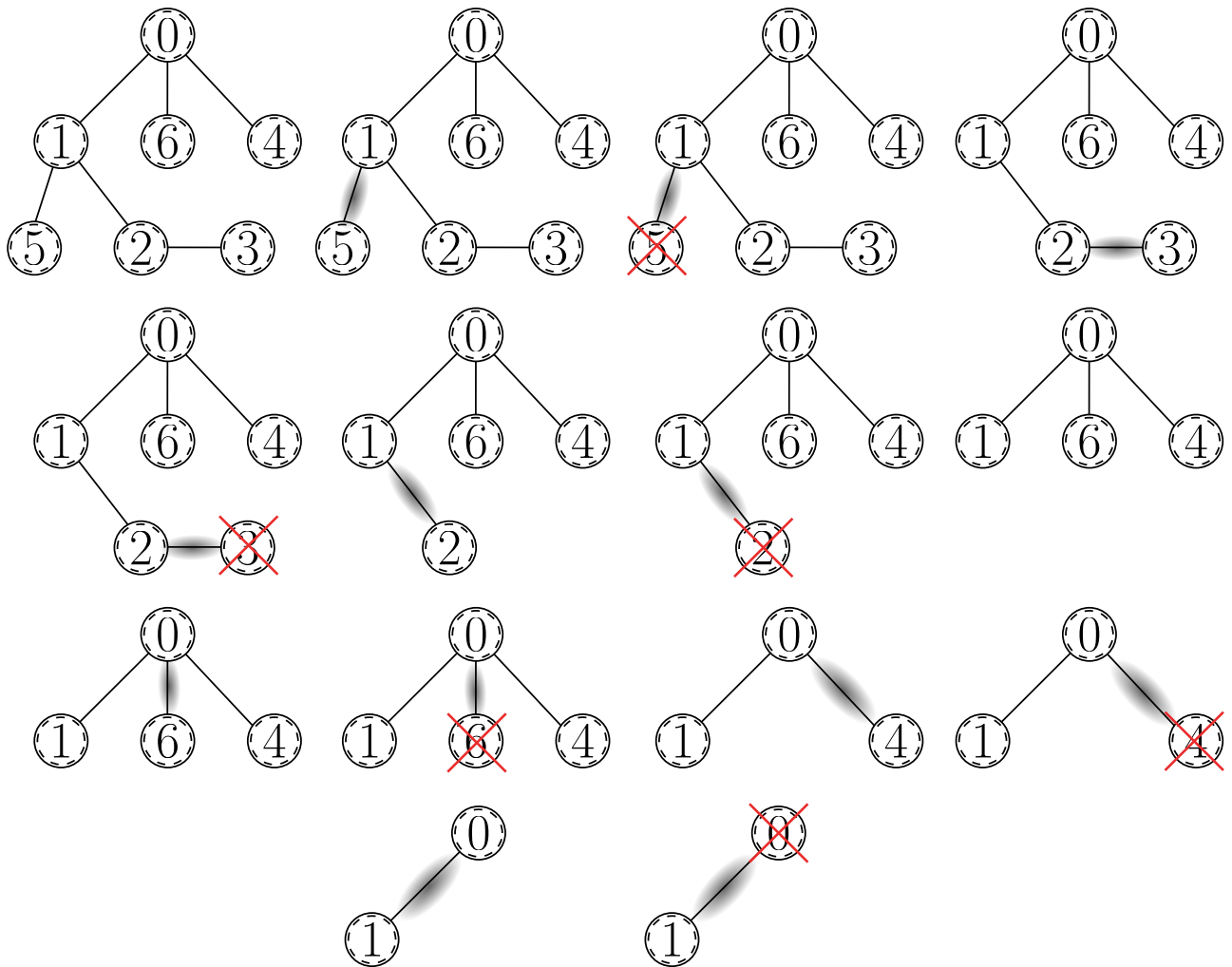
Para programas de C++, primero compila (es decir como `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) y ejecuta:

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

## Ejemplo

El ejemplo de esta sección fija  $N = 7$  por simplicidad y no es por lo tanto un caso válido. No se espera que tu programa sea capaz de resolver este caso. Todos los casos del grader tendrán  $N = 1000$ .

En el ejemplo, se le da a Ann el siguiente árbol. En la primera fase, Ann lee el árbol, elige una cadena binaria "0110" para enviar a Kathrin, y elige el orden  $[\ell_0, \ell_1, \dots, \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$  en el que las piezas se eliminan del árbol. En la segunda fase, Kathrin recibe la cadena "0110" enviada en la primera fase. Luego recibe la pareja (1, 5) y decide eliminar el vértice 5, el cual es una hoja. Para su siguiente movimiento, recibe la pareja (2, 3) y elimina la hoja 3, etc. Las siguientes imagenes ilustran las interacciones:



| salida del grader | tu salida |
|-------------------|-----------|
| 1 7               |           |
| 0 1               |           |
| 1 2               |           |
| 2 3               |           |
| 0 4               |           |
| 0 6               |           |
| 1 5               |           |
|                   | 0110      |
|                   | 5         |
|                   | 3         |
|                   | 2         |
|                   | 6         |
|                   | 4         |
|                   | 0         |

| salida del grader | tu salida |
|-------------------|-----------|
| 2 7               |           |
| 0110              |           |
| 1 5               |           |
|                   | 5         |
| 2 3               |           |
|                   | 3         |
| 1 2               |           |
|                   | 2         |
| 0 6               |           |
|                   | 6         |
| 0 4               |           |
|                   | 4         |
| 0 1               |           |
|                   | 0         |