D. Laser Strike

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

Ann y su amiga Kathrin acaban de descubrir un juego de mesa que se ha convertido en su favorito: Laser Strike. En este juego, las dos jugadoras trabajan en equipo para quitar N piezas del tablero. El juego se lleva a cabo en dos etapas. Lo que complica el juego es que Kathrin no tendrá la información completa del juego. Para poder ganar el juego, Ann y Kathrin tienen que trabajar juntas, mientras que se comunican lo menos posible.

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

En la primer etapa del juego, Ann escoge un orden $\ell_0, \ell_1, \ldots, \ell_{N-2}$ en el que se quitarán las piezas hasta que solo quede una. Este orden será secreto para Kathrin. Si puede replicarlo, ganarán el juego. Para quitar piezas se debe cumplir la siguiente regla: cada vez que se quita una pieza, debe estar conectada a exactamente una de las piezas restantes. En otras palabras, la pieza que se quita, debe ser una hoja del árbol formado por las piezas restantes y ella misma. (Después de que se hayan quitado $N-1$ piezas, la última pieza se quita automáticamente y las jugadoras ganan.) Ann debe escoger algún orden que cumpla la regla mencionada anteriormente.

Ann también va a escribir un mensaje para Kathrin, en la forma de una cadena binaria. Ann puede escoger qué tan largo es este mensaje – pero entre más corto sea el mensaje, obtendrán una mayor cantidad de puntos.

Después de eso, empieza la segunda etapa del juego. El objetivo del juego es que Kathrin quite $N-1$ piezas del tablero en el orden $\ell_0, \ell_1, \ldots, \ell_{N-2}$. Hará $N-1$ movimientos. Antes del movimiento i, Ann le dice a Kathrin un par de enteros a, b con las siguientes propiedades:

- $a < b$;
- todavía hay un par de piezas conectadas directamente con números a y b; y
- ya sea a o b es la pieza correcta ℓ_i que debe ser quitada en este turno.

Observa que para Ann la conexión (a, b) es determinada únicamente por la hoja ℓ_i del árbol actual.

Después Kathrin quita a o b del tablero. Si esta es la pieza correcta – o sea, ℓ_i – siguen jugando. De lo contrario, pierden el juego.

Tu tarea es implementar las estrategias de Ann y de Kathrin para que ganen el juego.

Los puntos que obtenga tu programa dependerán de la longitud del mensaje que Ann escriba en la primer etapa del juego.

Implementación

Tu programa será ejecutado exactamente dos veces. La primera vez que se ejecute, debe implementar la estrategia de Ann para la primer etapa del juego. Después, debe implementar la estrategia de Kathrin para la segunda etapa del juego.

La primer línea de la entrada contiene dos enteros P y N, donde P es 1 o 2 (primera o segunda etapa), y N es el número de piezas.

La siguiente parte de la entrada depende de la etapa del juego:

Etapa 1: Ann

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

Tu programa debe empezar imprimiendo una cadena binaria con a lo más $1\,000$ caracteres, cada uno debe ser `0` o `1`, el mensaje escrito por Ann. Observa que para generar una cadena de longitud 0, debes imprimir una línea vacía.

Después de esto, tu programa debe imprimir $N-1$ enteros $\ell_0, \ell_1, \ldots, \ell_{N-2}$ en líneas separadas, indicando el orden en el que Ann quiere quitar las hojas del árbol.

El orden debe ser tal que si las piezas se quitan del árbol de una por una en este orden, la pieza que se quita siempre debe ser una hoja, es decir, el árbol siempre tiene que permanecer conectado.

Etapa 2: Kathrin

Después de la primer línea (descrita anteriormente), la siguiente línea de la entrada contiene la cadena binaria (el mensaje de Ann) de la etapa 1.

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

En el i-ésimo movimiento, tu programa debe leer primero dos enteros a y b ($0 \leq a < b \leq N - 1$). Una de estas piezas es la hoja ℓ_i en el orden de Ann, y la otra pieza es la única pieza que queda que está conectada a ℓ_i. Después tu programa debe imprimir ℓ_i, indicando que Kathrin quitó esta hoja. Si tu programa no imprime la hoja ℓ_i correcta, las chicas pierden el juego y el veredicto de tu envío será Wrong Answer para este caso de prueba.

Detalles

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

Asegúrate de hacer flush del stream de salida después de cada línea que imprimas, de lo contrario tu programa podría obtener el veredicto Time Limit Exceeded. En Python, esto sucede automáticamente si usas `input()` para leer líneas. En C++, `cout << endl;` también hace flush además de imprimir una línea nueva; si usas printf deberás usar `fflush(stdout);`.

Ten en cuenta que leer una cadena vacía puede ser engañoso. Las plantillas proporcionadas manejan este caso correctamente.

Límites y Evaluación

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

Tu solución se evaluará con un conjunto de grupos de casos de prueba, cada grupo otorga un valor determinado de puntos. Cada grupo contiene un conjunto de casos de prueba. Para obtener los puntos de un grupo, tienes que resolver todos los casos de prueba de ese grupo.

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

Para cada grupo de prueba que tu programa resuelva correctamente, recibirás un puntaje basado en la siguiente fórmula:

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

donde S_g es el máximo puntaje para el grupo y K es la máxima longitud del mensaje de Ann que se necesita para cualquier caso de prueba en el grupo. **Tu puntaje para cada grupo será redondeado al próximo entero.**

La tabla de abajo muestra el número de puntos, para algunos valores de K, que tu programa obtendrá si resuelve todos los grupos con ese valor de K. Particularmente, para lograr un puntaje de 100 puntos en este problema, necesitas resolver todos los casos de prueba con $K \leq 1$.

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

Herramienta de Pruebas

Para facilitarte poder probar tu solución, te damos una herramienta simple que puedes descargar. Ve a la sección "attachments" al final de la página del problema en Kattis. El uso de la herramienta es opcional y puedes cambiarla. Ten en cuenta que el evaluador oficial en Kattis es diferente a esta herramienta de pruebas.

Para usar la herramienta, crea un archivo de entrada, como “sample1.in”, el cual debe empezar con un número N seguido de $N-1$ líneas especificando el árbol, en el mismo formato que en la etapa 1 del juego. Por ejemplo, para el caso de ejemplo que se ilustra más adelante:

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

Para programas en python, si tenemos `solution.py` (se corre normalmente `pypy3 solution.py`) corre:

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

Para programas en C++, primero compílalo (por ejemplo `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) y luego corre:

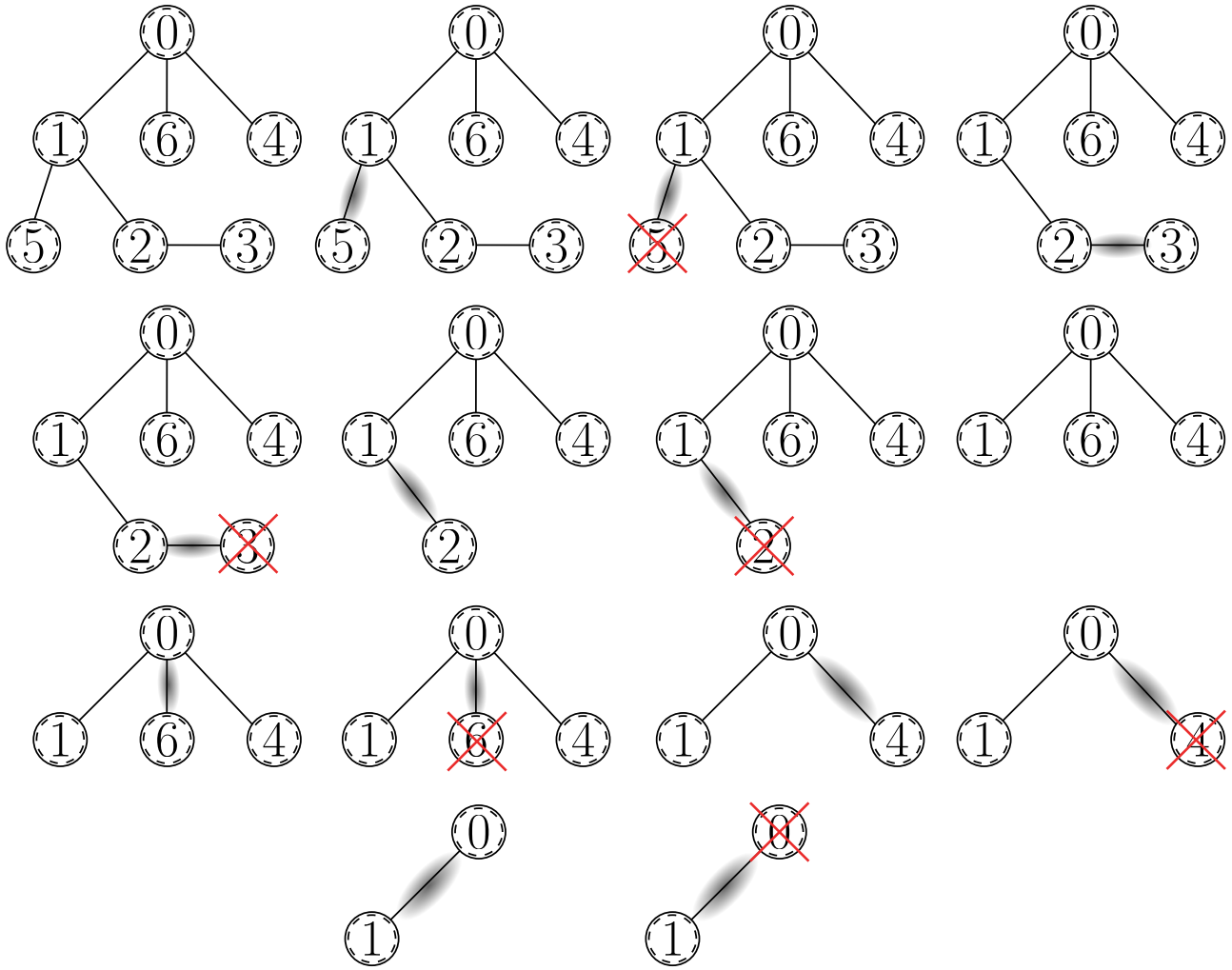
```
python3 testing_tool.py ./solution.out < sample1.in
```

Ejemplo

Observa que el ejemplo en esta sección tiene $N = 7$ por simplicidad y no es un caso de prueba válido. No se espera que tu programa resuelva el siguiente caso. Todos los casos de prueba en el evaluador tendrán $N = 1\,000$.

En este ejemplo, Ann tiene el siguiente árbol. En la primer etapa, Ann lee la descripción del árbol, elige la cadena binaria "`0110`" que le enviará a Kathrin, y elige el orden $[\ell_0, \ell_1, \ldots \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$ en el que se deben quitar las piezas.

En la segunda etapa, Kathrin recibe la cadena "`0110`" que le enviaron en la primera etapa. Ella entonces recibe el par $(1, 5)$ y decide quitar el vértice 5, que en efecto es una hoja. Para su siguiente movimiento, ella recibe el par $(2, 3)$ y quita la hoja 3, y así sucesivamente. Las siguientes imágenes muestran las interacciones del ejemplo:



salida del evaluador	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 evaluador	tu salida
2 7	
0110	
1 5	
	5
2 3	
	3
1 2	
	2
0 6	
	6
0 4	
	4
0 1	
	0