



El viaje más largo

¡Los organizadores de la IOI 2023 tienen un gran problema! Se olvidaron de planificar el viaje a Ópusztaszer para el día de mañana. Pero tal vez aún no sea demasiado tarde ...

Hay N puntos de referencia en Ópusztaszer numerados de 0 a $N - 1$. Algunas parejas de estos puntos de referencia están conectadas por **carreteras bidireccionales**. Cada pareja de puntos de referencia está conectada por una carretera como máximo. Los organizadores *desconocen* cuáles puntos de referencia están conectados por carreteras.

Decimos que la **densidad** de una red de carreteras de Ópusztaszer es **al menos** δ si cada 3 puntos de referencia distintos tienen al menos δ carreteras entre ellos. En otras palabras, para cada tripleta de puntos de referencia (u, v, w) tal que $0 \leq u < v < w < N$, entre las parejas de puntos de referencia (u, v) , (v, w) y (u, w) al menos δ parejas están conectadas por una carretera.

Los organizadores *conocen* un número entero positivo D tal que la densidad de la red de carreteras es al menos D . Ten en cuenta que el valor de D no puede ser mayor que 3.

Los organizadores pueden hacer **llamadas** al operador telefónico de Ópusztaszer para recopilar información sobre las conexiones por carretera entre determinados puntos de referencia. En cada llamada, se deben especificar dos arreglos no vacíos de puntos de referencia $[A[0], \dots, A[P - 1]]$ y $[B[0], \dots, B[R - 1]]$. Los puntos de referencia deben ser distintos por pares, es decir:

- $A[i] \neq A[j]$ para cada i y j tal que $0 \leq i < j < P$;
- $B[i] \neq B[j]$ para cada i y j tal que $0 \leq i < j < R$;
- $A[i] \neq B[j]$ para cada i y j tal que $0 \leq i < P$ y $0 \leq j < R$.

Para cada llamada, el despachador informa si hay una carretera que conecta un punto de referencia de A y un punto de referencia de B . Más precisamente, el despachador itera sobre todas las parejas i y j tal que $0 \leq i < P$ y $0 \leq j < R$. Si para alguna de ellas los puntos de referencia $A[i]$ y $B[j]$ están conectados por una carretera, el despachador retorna `true`. De lo contrario el despachador retorna `false`.

Un **viaje** de longitud l es una secuencia de puntos de referencia *distintos* $t[0], t[1], \dots, t[l - 1]$, donde para cada i entre 0 y $l - 2$ inclusivo, el punto de referencia $t[i]$ y el punto de referencia $t[i + 1]$ están conectados por una carretera. Un viaje de longitud l es llamado **viaje más largo** si no existe ningún viaje de longitud de al menos $l + 1$.

Tu tarea es ayudar a los organizadores a encontrar un viaje más largo en Ópusztaszer llamando al despachador.

Detalles de implementación

Deberás implementar la siguiente función:

```
int[] longest_trip(int N, int D)
```

- N : el número de puntos de referencia en Ópusztaszer.
- D : la densidad mínima garantizada de la red de carreteras.
- Esta función debe devolver un arreglo $t = [t[0], t[1], \dots, t[l-1]]$, representando un viaje más largo.
- Esta función se puede llamar **varias veces** para cada caso de prueba.

La función anterior puede realizar llamadas al siguiente procedimiento:

```
bool are_connected(int[] A, int[] B)
```

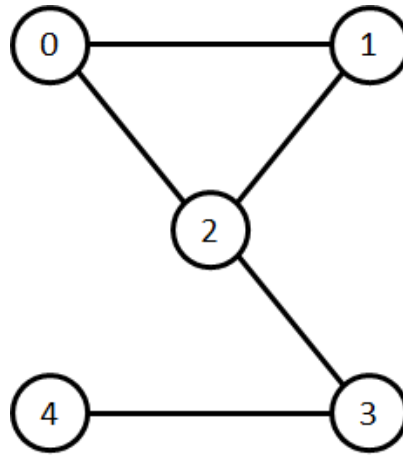
- A : un arreglo no vacío de puntos de referencia distintos.
- B : un arreglo no vacío de puntos de referencia distintos.
- A y B deben ser disjuntos.
- Este procedimiento retorna `true` si existe un punto de referencia de A y un punto de referencia de B conectados por una carretera. De lo contrario, retorna `false`.
- Este procedimiento se puede llamar como máximo 32 640 veces en cada invocación de `longest_trip`, y a lo más 150 000 veces en total.
- La longitud de los arreglos A y B pasadas a este procedimiento en todas sus invocaciones en total, no debe exceder 1 500 000.

El evaluador **no es adaptable**. Cada envío se califica dentro del mismo conjunto de casos de prueba. Es decir, los valores de N y D , así como las parejas de puntos de referencia conectados por carreteras, se fijan para cada llamada de `longest_trip` dentro de cada caso de prueba.

Ejemplos

Ejemplo 1

Considera un caso en el que $N = 5$, $D = 1$, y las conexiones por carretera son como se muestra en la siguiente figura:



La función `longest_trip` se llama de la siguiente forma:

```
longest_trip(5, 1)
```

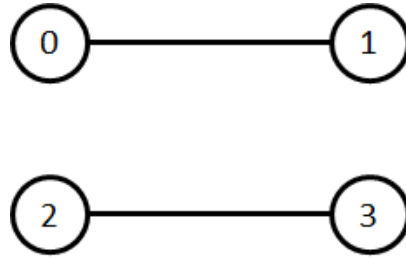
La función podría realizar llamadas a `are_connected` como sigue:

Llamada	Parejas conectadas por una carretera	Valor retornado
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) y (0,2)	true
<code>are_connected([2], [0])</code>	(2,0)	true
<code>are_connected([2], [3])</code>	(2,3)	true
<code>are_connected([1, 0], [4, 3])</code>	ninguno	false

Después de la cuarta llamada, resulta que *ninguna* de las parejas (1,4), (0,4), (1,3) y (0,3) están conectadas por una carretera. Como la densidad de la red es al menos $D = 1$, vemos que del triplete (0,3,4), la pareja (3,4) debe estar conectada por una carretera. De manera similar, los puntos de referencia 0 y 1 deben estar conectados.

Para este punto podemos concluir que $t = [1, 0, 2, 3, 4]$ es un viaje de longitud 5, y que no existe un viaje de longitud mayor que 5. Por lo tanto, la función `longest_trip` puede devolver `[1, 0, 2, 3, 4]`.

Considera otro caso en el que $N = 4$, $D = 1$ y las carreteras entre los puntos de referencia son como se muestra en la siguiente figura:



La función `longest_trip` es llamada de la siguiente forma:

```
longest_trip(4, 1)
```

En este caso, el tamaño del viaje más largo es 2. Por lo tanto, después de algunas llamadas al procedimiento `are_connected`, la función `longest_trip` debe devolver alguno de los siguientes: $[0, 1]$, $[1, 0]$, $[2, 3]$ o $[3, 2]$.

Ejemplo 2

La subtarea 0 contiene un caso de prueba de ejemplo adicional con $N = 256$ puntos de referencia. Este caso de prueba está incluido en el archivo adjunto que puedes descargar desde el sistema del concurso.

Límites

- $3 \leq N \leq 256$
- La suma de N sobre todas las llamadas a `longest_trip` no excederá 1 024 en cada caso de prueba.
- $1 \leq D \leq 3$

Subtareas

1. (5 puntos) $D = 3$
2. (10 puntos) $D = 2$
3. (25 puntos) $D = 1$. Sea l^* el tamaño de un viaje más largo. La función `longest_trip` no tiene que retornar un viaje de longitud l^* . En cambio, debería devolver un viaje de al menos $\left\lceil \frac{l^*}{2} \right\rceil$.
4. (60 puntos) $D = 1$

En la subtarea 4, tu puntaje se determina en función del número de llamadas al procedimiento `are_connected` durante una única invocación de `longest_trip`. Sea q el número máximo de llamadas entre todas las invocaciones de `longest_trip` en cada caso de prueba de la subtarea. Tu puntuación para esta subtarea se calcula según la siguiente tabla:

Condición	Puntos
$2\,750 < q \leq 32\,640$	20
$550 < q \leq 2\,750$	30
$400 < q \leq 550$	45
$q \leq 400$	60

Si en cualquiera de los casos de prueba las llamadas al procedimiento `are_connected` no cumplen con las restricciones descritas en detalles de implementación, o el arreglo devuelto por `longest_trip` es incorrecto, la puntuación de tu solución para esa subtask será 0.

Evaluador de ejemplo

Sea C el número de casos, es decir, el número de llamadas a `longest_trip`. El evaluador de ejemplo lee la entrada en el siguiente formato:

- línea 1: C

Seguido de las descripciones de los C casos.

El evaluador de ejemplo lee la descripción de cada caso en el siguiente formato:

- línea 1: $N\ D$
- línea $1 + i$ ($1 \leq i < N$): $U_i[0]\ U_i[1]\ \dots\ U_i[i - 1]$

Aquí cada U_i ($1 \leq i < N$) es un arreglo de tamaño i , describiendo cuales parejas de puntos de referencia están conectadas por una carretera. Para cada i y j tal que $1 \leq i < N$ y $0 \leq j < i$:

- Si los puntos de referencia j y i estan conectados por una carretera, entonces el valor de $U_i[j]$ debería ser 1;
- Si no hay una carretera que conecte los puntos de referencia j y i , entonces el valor de $U_i[j]$ debería ser 0.

En cada caso, antes de llamar `longest_trip`, el evaluador de ejemplo verifica si la densidad de la red de carreteras es al menos D . Si esta condición no se cumple, imprime el mensaje `Insufficient Density` y termina.

Si el evaluador de ejemplo detecta una violación del protocolo, la salida del evaluador de ejemplo es `Protocol Violation: <MSG>`, donde `<MSG>` es uno de los siguientes mensajes de error:

- `invalid array`: en una llamada a `are_connected`, al menos uno de los arreglos A y B
 - está vacío, o
 - contiene un elemento que no es un entero entre 0 y $N - 1$, inclusivo, o
 - contiene el mismo elemento al menos dos veces.

- `non-disjoint arrays`: en una llamada a `are_connected`, los arreglos A y B no son disjuntos.
- `too many calls`: el número de llamadas realizadas a `are_connected` excede 32 640 en la actual invocación de `longest_trip`, o excede 150 000 en total.
- `too many elements`: el número total de puntos de referencia pasados a `are_connected` sobre todas las llamadas excede 1 500 000.

De lo contrario, supón que los elementos del arreglo devuelto por `longest_trip` en cada caso serán $t[0], t[1], \dots, t[l-1]$ para algún l no negativo. El evaluador de ejemplo imprimirá tres líneas para ese caso en el siguiente formato:

- línea 1: l
- línea 2: $t[0] \ t[1] \ \dots \ t[l-1]$
- línea 3: el número de llamadas a `are_connected` en este caso

Finalmente, el evaluador de ejemplo imprime:

- línea $1 + 3 \cdot C$: el número máximo de llamadas a `are_connected` sobre todas las llamadas a `longest_trip`.