



Tren de Juguete

Arezou y su hermano Borzou son mellizos. Recibieron un fabuloso kit de trenes de juguete para su cumpleaños, y lo utilizaron para construir una red de ferrocarril con n estaciones y m tramos de vía *dirigidos*. Las estaciones están numeradas desde 0 hasta $n - 1$. Cada tramo comienza en una estación, y termina, o bien en esa misma estación o en otra diferente. En cada estación comienza por lo menos un tramo de vía.

Algunas estaciones son *estaciones de recarga*. Cada vez que el tren llega a una estación de recarga, se recarga por completo. Un completamente cargado tiene suficiente energía para recorrer n tramos consecutivos. Es decir, justo al entrar al tramo $n + 1$ desde su última recarga, el tren se queda sin energía y se detiene.

En cada estación hay un mecanismo de selección, que se puede hacer apuntar a cualquiera de los tramos que comienzan en esa estación. El tren siempre sale de la estación utilizando el tramo de vía apuntado por el mecanismo.

Los mellizos van a jugar un juego con su tren. Ya se han repartido todas las estaciones entre ellos, de manera que cada una pertenece o bien a Arezou o bien a Borzou. Hay un único tren en el juego. Al comienzo del juego, el tren se encuentra en la estación s y está completamente cargado. Para comenzar el juego, el dueño de la estación s apunta el mecanismo en la estación s a uno de los tramos salientes. Luego encienden el tren, y este comienza a moverse por las vías.

Cada vez que el tren entra a una estación por primera vez, el dueño de esa estación apunta el mecanismo de esa estación a una de sus vías salientes. Una vez que el mecanismo de una estación es apuntado, se queda en la misma posición por el resto del juego. Por lo tanto, si un tren reingresa a una estación que ya visitó anteriormente, saldrá de la misma por exactamente el mismo tramo de vía por el que lo hizo la primera vez.

Como hay un número finito de estaciones, eventualmente el tren ingresará en un *ciclo*. Un ciclo es una secuencia de estaciones *distintas* $c[0], c[1], \dots, c[k - 1]$ tales que para cada $0 \leq i < k - 1$, el tren sale de la estación $c[i]$ por un tramo de vía que llega a la estación $c[i + 1]$, y además el tren sale de la estación $c[k - 1]$ por un tramo de vía que llega a la estación $c[0]$. Un ciclo puede tener una única estación (siendo $k = 1$) si el tren sale de la estación $c[0]$ usando un tramo de vía que vuelve a $c[0]$.

Arezou gana el juego si el tren continua viajando indefinidamente, y Borzou gana si el tren se queda sin energía. En otras palabras, si hay al menos una estación de recarga entre $c[0], c[1], \dots, c[k - 1]$, el tren puede recargar y ciclar indefinidamente, y Arezou gana. De lo contrario, se quedará sin energía (posiblemente luego de ciclar varias veces) y Borzou gana.

Dada la descripción de la red de ferrocarril, Arezou y Borzou jugarán n juegos. En el juego s , para $0 \leq s \leq n - 1$, el tren se encontrará inicialmente en la estación s . Tu tarea consiste en decidir, para cada juego, si existe una estrategia para Arezou que le garantice ganar sin importar lo que Borzou juegue.

Detalles de implementación

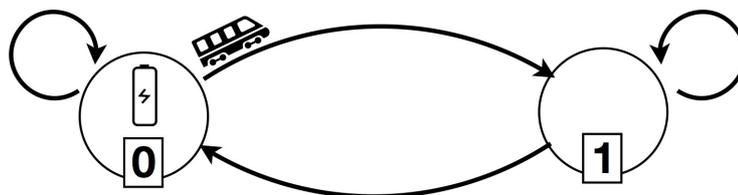
Debes implementar la siguiente función:

```
int[] who_wins(int[] a, int[] r, int[] u, int[] v)
```

- a : arreglo de longitud n . Si Arezou es la dueña de la estación i , $a[i] = 1$. De lo contrario, Borzou es dueño de la estación i y se tiene $a[i] = 0$.
- r : arreglo de longitud n . Si la estación i es una estación de recarga, $r[i] = 1$. De lo contrario, $r[i] = 0$.
- u y v : arreglos de longitud m . Para todo $0 \leq i \leq m - 1$, hay un tramo de vía dirigido desde la estación $u[i]$ hasta la estación $v[i]$.
- La función debe retornar un arreglo w de longitud n . Para cada $0 \leq i \leq n - 1$, el valor de $w[i]$ debe ser 1 si Arezou gana el juego que comienza en la estación i , sin importar cómo juegue Borzou. De lo contrario, el valor de $w[i]$ deberá ser 0.

Ejemplo

```
who_wins([0, 1], [1, 0], [0, 0, 1, 1], [0, 1, 0, 1])
```



- Hay 2 estaciones. Borzou es el dueño de la estación 0, que es una estación de recarga. Arezou es la dueña de la estación 1, que no es una estación de recarga.
- Hay 4 tramos $(0, 0)$, $(0, 1)$, $(1, 0)$ y $(1, 1)$, donde (i, j) denota un tramo de vía dirigido desde la estación i hasta la estación j .
- Consideremos el juego en el que el tren se ubica inicialmente en la estación 0. Si Borzou apunta el mecanismo de la estación 0 hacia el tramo $(0, 0)$, el tren ciclará indefinidamente por este tramo (notar que la estación 0 es una estación de recarga). En este caso, Arezou gana. En cambio, si Borzou apunta el mecanismo de la estación 0 hacia el tramo $(0, 1)$, Arezou puede apuntar el mecanismo de la estación 1 hacia $(1, 0)$. Si esto ocurre, el tren ciclará indefinidamente entre ambas estaciones. Nuevamente Arezou gana, ya que la estación 0 es una estación de recarga, y el tren no se detendrá. Luego, Arezou puede ganar el juego, sin importar lo que haga Borzou.

- Siguiendo un razonamiento similar, en el juego que comienza en la estación 1, Arezou también gana sin importar como juegue Borzou. Por lo tanto, la función debe retornar $[1, 1]$.

Restricciones

- $1 \leq n \leq 5000$.
- $n \leq m \leq 20\,000$.
- Hay al menos una estación de recarga.
- Hay al menos un tramo de vía comenzando en cada estación.
- Podría haber tramos de vía que comiencen y terminen en la misma estación ($u[i] = v[i]$).
- Cada tramo de vía es diferente. Es decir, no hay dos índices i y j con $0 \leq i < j \leq m - 1$ tales que $u[i] = u[j]$ y $v[i] = v[j]$.
- $0 \leq u[i], v[i] \leq n - 1$ (para todo $0 \leq i \leq m - 1$).

Subtareas

1. (5 puntos) Para todo $0 \leq i \leq m - 1$, o bien $v[i] = u[i]$ o bien $v[i] = u[i] + 1$.
2. (10 puntos) $n \leq 15$.
3. (11 puntos) Arezou es la dueña de todas las estaciones.
4. (11 puntos) Borzou es el dueño de todas las estaciones.
5. (12 puntos) Hay exactamente una estación de recarga.
6. (51 puntos) Sin restricciones adicionales.

Evaluador de ejemplo

El evaluador de ejemplo lee la entrada con el siguiente formato:

- línea 1: $n \ m$
- línea 2: $a[0] \ a[1] \ \dots \ a[n - 1]$
- línea 3: $r[0] \ r[1] \ \dots \ r[n - 1]$
- línea 4 + i (para $0 \leq i \leq m - 1$): $u[i] \ v[i]$

El evaluador de ejemplo imprime el valor de retorno de `who_wins` en el siguiente formato:

- línea 1: $w[0] \ w[1] \ \dots \ w[n - 1]$