



Tren de juguete

Arezou y su hermano Borzou son gemelos. Ellos han recibido un increíble juego de trenes para su cumpleaños y lo usaron para construir un sistema de vías de tren con n estaciones y m vías que tienen un solo sentido. Las estaciones están numeradas de 0 a $n - 1$. Cada vía de tren inicia en una estación y termina en la misma estación o a alguna otra estación distinta. Al menos hay una vía de tren que inicia en cada estación.

Algunas estaciones son *estaciones de carga*. Cuando un tren parte de una estación de carga, lo hace con la carga llena. Un tren con la carga llena tiene suficiente energía para viajar a través de n vías de tren consecutivas. Es decir, el tren se queda sin energía cuando entra en la estación $n + 1$ si no fue cargado.

En cada estación existe un *interruptor* que apunta a alguna de las vías de tren que inician en esta estación. Cuando un tren sale de una estación, lo hace por la vía de tren a la que el *interruptor* está apuntando.

Los gemelos tienen un juego con el tren. Se dividieron las estaciones entre ellos: cada estación pertenece a Arezou o a Borzou. Sólo hay un tren. Al inicio del juego el tren se encuentra en la estación s con la carga llena. Al iniciar el juego, el dueño de la estación s apunta el *interruptor* de la estación s a alguna de las vías de tren que inician en la estación s . Después, ellos prenden el tren y el tren empezará a viajar por las vías.

Cada que el tren entra a una estación por primera vez, el dueño de la estación acomoda el *interruptor* de esa estación. Una vez que el *interruptor* esté acomodado, se quedará en esa posición por el resto del juego. Por lo tanto, si un tren vuelve a entrar a una estación visitada anteriormente, partirá por la misma vía de tren por la que partió la primera vez que visitó esa estación.

Dado que hay una cantidad finita de estaciones, el tren eventualmente entrará en un *ciclo*. Un *ciclo* es una secuencia de estaciones distintas $c[0], c[1], \dots, c[k - 1]$ tal que el tren abandona la estación $c[i]$, (para $0 \leq i \leq k - 1$) usando una vía de tren con dirección a la estación $c[i + 1]$. De forma similar, el tren deja la estación $c[k - 1]$ con dirección a la estación $c[0]$. Un ciclo puede tener una sola estación (i.e $k = 1$) si la vía de tren que parte de la estación $c[0]$, regresa a la misma estación $c[0]$.

Arezou gana el juego si el tren continúa en movimiento por siempre y Borzou gana el juego si el tren se queda sin energía. En otras palabras, si hay al menos una estación de carga en un ciclo $c[0], c[1], \dots, c[k - 1]$, el tren puede recargarse indefinidamente y Arezou gana el juego. De la otra forma, el tren se queda sin energía (posiblemente después de completar el ciclo varias veces) y Borzou gana.

Se te proporciona la descripción del sistema de vías. Arezou y Borzou van a jugar n juegos. En el s -ésimo juego, [para $0 \leq s \leq n - 1$], el tren iniciará en la estación s . Tu tarea es encontrar, para cada juego, si existe una estrategia en la que Arezou está garantizado que gane, sin importar cómo Borzou juegue.

Detalles de implementación

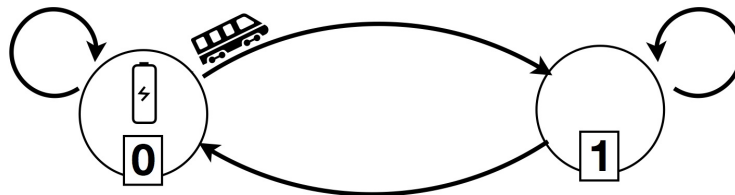
Tienes que implementar el siguiente procedimiento:

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

- a : arreglo de longitud n . Si Arezou es dueño de la estación i , entonces $a[i] = 1$. De lo contrario, si Borzou es dueño de la estación i , entonces $a[i] = 0$.
- r : arreglo de longitud n . Si la estación i es una estación de carga, entonces $r[i] = 1$. De lo contrario, $r[i] = 0$.
- u y v : arreglos de longitud m . Para todo $0 \leq i \leq m - 1$, existe una vía de tren que inicia en la estación $u[i]$ y termina en la estación $v[i]$.
- El procedimiento debe regresar un arreglo w de longitud n . Para cada $0 \leq i \leq n - 1$, el valor $w[i]$ debe ser 1 si Arezou puede ganar el juego cuando el tren inicia en la estación i , sin importar cómo Borzou juegue. De lo contrario, el valor de $w[i]$ debe ser 0.

Ejemplo

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



- Existen 2 estaciones. Borzou es el dueño de la estación 0. Esta estación es una estación de carga. Arezou es el dueño de la estación 1. Esta estación *no* es una estación de carga.
- Existen 4 vías de tren: $(0, 0)$, $(0, 1)$, $(1, 0)$, y $(1, 1)$, donde (i, j) denota una vía de tren de un sólo sentido que inicia en la estación i y termina en la estación j .
- Considera el juego en el que el tren está inicialmente en la estación 0. Si Borzou apunta el interruptor de la estación 0 hacia la vía de tren $(0, 0)$, entonces el tren entrará en un ciclo infinitamente (nota que la estación 0 es la estación de carga). En este caso, Arezou gana. De otra forma, si Borzou asigna el interruptor de la estación 0 hacia la vía de tren $(0, 1)$, Arezou puede asignar el interruptor de la estación 1 hacia la vía de tren $(1, 0)$. Si esto pasa, el tren entrará en un ciclo entre las dos estaciones. De nuevo Arezou gana, pues la estación 0 es una estación de carga y el tren no se detendrá. Por lo tanto, Arezou puede ganar el juego, sin importar lo que Borzou haga.

- De forma similar, si el juego inicia en la estación 1 Arezou puede ganar, sin importar como Borzou juegue. Por lo tanto, el procedimiento debe regresar $[1, 1]$.

Restricciones

- $1 \leq n \leq 5000$.
- $n \leq m \leq 20\,000$.
- Existe al menos una estación de carga.
- Existe al menos una vía de tren que inicia en cada estación.
- Pueden existir vías de tren que inicien y terminen en la misma estación (i.e., $u[i] = v[i]$).
- Cada vía de tren es distinta. En otras palabras, no hay dos índices i y j ($0 \leq i < j \leq m - 1$) tal 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 $0 \leq i \leq m - 1$, se cumplirá que $v[i] = u[i]$ o $v[i] = u[i] + 1$.
2. (10 puntos) $n \leq 15$.
3. (11 puntos) Arezou es dueño de todas las estaciones.
4. (11 puntos) Borzou es dueño de todas las estaciones.
5. (12 puntos) Existe exactamente una estación.
6. (51 puntos) No hay restricciones adicionales.

Evaluador de ejemplo

El evaluador de ejemplo lee la entrada en 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 lo que regresa `who_wins` en el siguiente formato:

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