Spielzeugeisenbahn

Arezou und ihr Bruder Borzou sind Zwillinge. Sie haben zum Geburtstag eine tolle Spielzeugeisenbahn bekommen und haben damit ein Eisenbahnsystem mit n Stationen und m in nur eine Richtung befahrbaren Gleisen gebaut. Die Stationen sind von 0 bis $n-1$ durchnummeriert. Jedes Gleis beginnt an einer Station und endet an derselben oder an einer anderen Station. An jeder Station beginnt mindestens ein Gleis.

Manche Stationen sind *Aufladestationen*. Immer wenn der Zug an einer Aufladestation ankommt, wird er voll aufgeladen. Ein voll aufgeladener Zug hat genug Energie, um über n aufeinanderfolgende Gleise zu fahren. Das heißt, die Energie geht aus, sobald der Zug in das $(n+1)$-te Gleis nach dem letzten Aufladen einfährt.

An jeder Station befindet sich eine Weiche, die auf ein beliebiges der Gleise, die an der Station beginnen, gestellt werden kann. Wenn der Zug an einer Station ist, verlässt er sie über das Gleis, auf das die Weiche an dieser Station gestellt ist.

Die Zwillinge wollen ein Spiel mit ihrem Zug spielen. Sie haben schon alle Stationen unter sich aufgeteilt: Jede Station gehört entweder Arezou oder Borzou. Es gibt genau einen Zug. Am Anfang des Spiels befindet sich der Zug an Station s und ist voll aufgeladen. Um das Spiel zu beginnen, stellt der Besitzer der Station s die Weiche bei Station s auf eines der Gleise, die bei Station s beginnen. Dann schalten sie den Zug ein, und der Zug fährt los.

Immer wenn der Zug zum ersten Mal an einer Station ankommt, stellt der Besitzer dieser Station die Weiche an dieser Station. Wenn die Weiche einmal gestellt ist, bleibt sie bis zum Ende des Spiels in derselben Stellung. Wenn der Zug daher ein zweites Mal an einer Station ankommt, die er bereits zuvor durchfahren hat, verlässt er die Station wieder über dasselbe Gleis.

Da es nur endlich viele Stationen gibt, wird der Zug früher oder später anfangen, im Kreis zu fahren. Ein *Kreis* ist eine Folge von *verschiedenen* Stationen $c[0], c[1], \cdots, c[k-1]$, sodass der Zug die Station $c[i]$ (für $0 \leq i < k-1$) über ein Gleis verlässt, das zu Station $c[i+1]$ fährt, und die Station $c[k-1]$ über ein Gleis, das zu Station $c[0]$ fährt. Beachte, dass ein Kreis aus einer einzelnen Station bestehen kann (d.h. $k = 1$), wenn der Zug die Station $c[0]$ über ein Gleis verlässt, das wieder zurück zu $c[0]$ fährt.

Arezou gewinnt das Spiel, wenn der Zug unbegrenzt lange weiterfährt, und Borzou gewinnt, wenn dem Zug die Energie ausgeht. Mit anderen Worten, wenn sich unter den Stationen $c[0], c[1], \cdots, c[k-1]$ mindestens eine Ladestation befindet, kann der Zug aufgeladen werden und unbegrenzt lange im Kreis fahren, und Arezou gewinnt. Sonst geht dem Zug die Energie aus (möglicherweise nachdem er mehrmals im Kreis gefahren ist), und Borzou gewinnt.

Du bekommst die Beschreibung des Eisenbahnsystems. Arezou und Borzou werden n Spiele spielen. Im s-ten Spiel, für $0 \leq s \leq n-1$, befindet sich der Zug am Anfang an Station s. Deine Aufgabe ist es, für jedes Spiel herauszufinden, ob es eine Strategie für Arezou gibt, die garantiert, dass sie gewinnt, egal wie Borzou spielt.

Implementierung

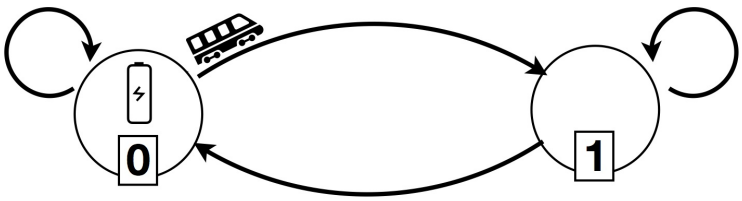
Implementiere die folgende Funktion:

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

- a: Ein Array der Länge n. Falls Station i Arezou gehört, ist $a[i] = 1$. Ansonsten gehört Station i Borzou und $a[i] = 0$.
- r: Ein Array der Länge n. Falls die Station i eine Aufladestation ist, ist $r[i] = 1$, ansonsten ist $r[i] = 0$.
- u und v: Arrays der Länge m. Für jedes i mit $0 \leq i \leq m-1$ gibt es ein (nur in eine Richtung befahrbares) Gleis, das an Station $u[i]$ beginnt und bei Station $v[i]$ endet.
- Die Funktion soll ein Array w der Länge n zurückgeben. Für jedes i mit $0 \leq i \leq n-1$ gilt: $w[i] = 1$, falls Arezou das Spiel gewinnen kann, in dem der Zug zu Beginn an Station i steht, egal was Borzou tut. Ansonsten ist $w[i] = 0$.

Beispiel

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



- Es gibt 2 Stationen. Station 0 gehört Borzou und ist eine Aufladestation. Station 1 gehört Arezou und ist keine Aufladestation.
- Es gibt 4 Gleise $(0, 0), (0, 1), (1, 0)$ und $(1, 1)$. (i, j) steht für ein Gleis von Station i nach Station j.
- Betrachte das Spiel, in dem der Zug zu Beginn an Station 0 steht. Falls Borzou die Weiche an Station 0 auf Gleis $(0, 0)$ stellt, wird der Zug unbegrenzt lange über dieses Gleis im Kreis fahren (beachte: 0 ist eine Aufladestation). In diesem Fall gewinnt Arezou. Ansonsten, falls Borzou die Weiche an Station 0 auf Gleis $(0, 1)$ setzt, kann Arezou die Weiche an Station 1 auf $(1, 0)$ setzen. Dann fährt der Zug unbegrenzt lange über beide Stationen im Kreis. Auch hier gewinnt Arezou, weil Station 0 eine Aufladestation ist und der Zug nicht anhält. Also kann Arezou das Spiel gewinnen, egal was Borzou tut.
- Auch das Spiel, in dem der Zug zu Beginn an Station 1 steht, kann Arezou gewinnen, egal was

Borzou tut. Die Begründung ist ähnlich wie oben.
- Die Funktion sollte deshalb $[1, 1]$ zurückgeben.

Beschränkungen

- $1 \leq n \leq 5000$.
- $n \leq m \leq 20\,000$.
- Es gibt mindestens eine Aufladestation.
- An jeder Station beginnt mindestens ein Gleis.
- Es kann Gleise geben, die an der gleichen Station enden, an der sie beginnen (d.h. $u[i] = v[i]$).
- Von einer Station i zu einer anderen Station j führt höchstens ein Gleis. Formal: Es gibt keine zwei Indizes i und j ($0 \leq i < j \leq m - 1$) so dass $u[i] = u[j]$ und $v[i] = v[j]$.
- $0 \leq u[i], v[i] \leq n - 1$ (für alle $0 \leq i \leq m - 1$).

Subtasks

1. (5 Punkte) Für alle $0 \leq i \leq m - 1$ ist entweder $v[i] = u[i]$ oder $v[i] = u[i] + 1$.
2. (10 Punkte) $n \leq 15$.
3. (11 Punkte) Alle Stationen gehören Arezou.
4. (11 Punkte) Alle Stationen gehören Borzou.
5. (12 Punkte) Es gibt genau eine Aufladestation.
6. (51 Punkte) Keine weiteren Beschränkungen.

Beispiel-Grader

Der Beispiel-Grader liest die Eingaben in folgendem Format:

- Zeile 1: n m
- Zeile 2: $a[0]$ $a[1]$ $\ldots$ $a[n-1]$
- Zeile 3: $r[0]$ $r[1]$ $\ldots$ $r[n-1]$
- Zeile $4 + i$ (für $0 \leq i \leq m - 1$): $u[i]$ $v[i]$

Der Beispiel-Grader gibt den Rückgabewert von `who_wins` in folgendem Format aus:

- Zeile 1: $w[0]$ $w[1]$ $\ldots$ $w[n-1]$