Vláčky

Arezou a její bratr Borzou jsou dvojčata. Dostali k narozeninám úžasnou stavebnici a použili ji k vybudování modelového kolejiště s n stanicemi číslovanými 0 až $n-1$ a m *jednosměrnými* kolejemi. Každá kolej začíná v jedné stanici a končí ve stejné nebo jiné stanici. V každé stanici začíná nejméně jedna kolej.

Některé stanice slouží jako *nabíjecí*. Kdykoli vláček dorazí do nabíjecí stanice, plně se nabije. Plně nabitý vláček má dostatek energie k projetí n na sebe navazujících kolejí. To znamená, že kdyby vláček chtěl vyjet na $(n+1)$. kolej od posledního dobití, nemá už energii a zastaví se.

Na každé stanici je výhybka, která může nasměrovat vláček na libovolnou kolej vedoucí z této stanice. Vláček opouští stanici po té koleji, kam byla v této stanici nastavena výhybka.

Dvojčata si budou s vláčkem hrát. Rozdělila si všechny stanice mezi sebou: každou stanici vlastní buď Arezou nebo Borzou. Ve hře účinkuje jediný vláček. Na začátku hry je vláček ve stanici s a je plně nabitý.

Vlastník stanice s nastaví z této stanici výhybku na některou z odsud vycházejících kolejí. Nastartuje vláček a ten se začne pohybovat po kolejišti.

Kdykoli vláček poprvé vjede do určité stanice, vlastník této stanice nastaví výhybku v této stanici. Výhybka poté zůstane stejně nastavena až do konce hry. Tudíž vjede-li vláček do již navštívené stanice, opustí ji po téže koleji jako při předchozí návštěvě.

Jelikož počet stanic je konečný, dříve či později se vláček dostane do *cyklu*. Cyklus je posloupnost *různých* stanic $c[0], c[1], \cdots, c[k-1]$ taková, že vláček ze stanice $c[i]$ (pro $0 \le i < k$) jede po koleji do stanice $c[i+1]$ a ze stanice $c[k-1]$ jede do stanice $c[0]$. Cyklus může mít jen jednu stanici, tj. $k=1$, jestliže existuje kolej vedoucí ze stanice $c[0]$ zpět do $c[0]$.

Arezou vyhraje, jestliže vláček bude jezdit donekonečna, zatímco Borzou zvítězí, když vláčku dojde energie.

Jinými slovy, je-li mezi stanicemi $c[0], c[1], \cdots, c[k-1]$ nějaká nabíjecí, vláček se dobije, pokračuje takto donekonečna a Arezou vyhraje. V opačném případě mu dojde energie (případně po vícenásobném projetí cyklu) a vyhraje Borzou.

Máte dánu reprezentaci kolejiště. Arezou a Borzou budou hrát n her. V s-té hře (pro $0 \le s \le n-1$) bude vláček začínat ve stanici s. Vaším úkolem pro každou hru je rozhodnout, zda existuje strategie pro Arezou, která jí zajistí výhru bez ohledu na to, jak bude hrát Borzou.

Podrobnosti k implementaci

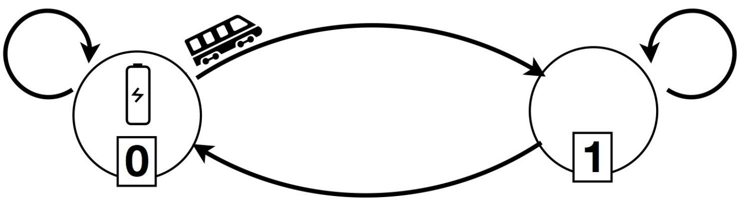
Máte za úkol implementovat následující proceduru:

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

- a: pole délky n. Jestliže Arezou vlastní stanici i, je $a[i] = 1$. Jinak stanici i vlastní Borzou a $a[i] = 0$.
- r: pole délky n. Jestliže je stanice i nabíjecí, je $r[i] = 1$. Jestliže ne, pak $r[i] = 0$.
- u a v: pole délky m. Pro všechna $0 \leq i \leq m - 1$ existuje jednosměrná kolej ze stanice $u[i]$ do stanice $v[i]$.
- Procedura musí vrátit pole w délky n. Pro každé $0 \leq i \leq n - 1$ musí být hodnota $w[i]$ rovna 1, jestliže hru začínající ve stanici i vyhraje Arezou bez ohledu na to, jak bude hrát Borzou. Jinak musí být $w[i]$ rovno 0.

Příklad

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



- V příkladu jsou 2 stanice. Borzou je vlastníkem stanice 0, což je zároveň nabíjecí stanice. Arezou je vlastníkem stanice 1, která nabíjecí stanicí není.
- Existují 4 koleje $(0, 0), (0, 1), (1, 0)$ a $(1, 1)$, kde (i, j) označuje jednosměrnou kolej ze stanice i do stanice j.
- Uvažme hru, v níž je vláček zpočátku ve stanici 0. Pokud Borzou nastaví výhybku ve stanici 0 na kolej $(0, 0)$, bude vláček po této koleji donekonečna jezdit díky tomu, že stanice 0 je nabíjecí. V tomto případě vyhrála Arezou. V opačném případě, pokud Borzou nastaví výhybku ve stanici 0 na kolej $(0, 1)$, může Arezou nastavit výhybku ve stanici 1 na kolej $(1, 0)$. Bude-li tomu tak, bude vláček donekonečna projíždět mezi oběma stanicemi a opět vyhraje Arezou, protože stanice 0 je nabíjecí a vláček se nezastaví. Arezou tedy může hru vyhrát bez ohledu na to, co dělá Borzou.
- Podobnou úvahou dospějeme k tomu, že Arezou vyhraje i hru, kde je vláček na začátku ve stanici 1.
- Jelikož v obou hrách vyhraje Arezou, procedura by měla vrátit $[1, 1]$.

Omezení

- $1 \leq n \leq 5000$.
- $n \leq m \leq 20\,000$.
- V kolejišti je alespoň jedna nabíjecí stanice.

- V každé stanici začíná aspoň jedna kolej.
- Mohou existovat koleje vedoucí ze stanice do ní samé (tj. $u[i] = v[i]$).
- Všechny koleje jsou unikátní (nejsou dvě stejné), jinak řečeno neexistují indexy i a j ($0 \leq i < j \leq m-1$) takové, že $u[i] = u[j]$ a $v[i] = v[j]$.
- $0 \leq u[i], v[i] \leq n-1$ (pro všechna $0 \leq i \leq m-1$).

Podúlohy

1. (5 bodů) Pro všechna $0 \leq i \leq m-1$ platí buď $v[i] = u[i]$ nebo $v[i] = u[i] + 1$.
2. (10 bodů) $n \leq 15$.
3. (11 bodů) Arezou vlastní všechny stanice.
4. (11 bodů) Borzou vlastní všechny stanice.
5. (12 bodů) V kolejišti je právě jedna nabíjecí stanice.
6. (51 bodů) Žádné dodatečné podmínky.

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač čte vstup v následujícím formátu:

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

Ukázkový vyhodnocovač vypíše návratovou hodnotu procedury `who_wins` v následujícím formátu:

- řádek 1: $\;w[0] \;\; w[1] \;\ldots\; w[n-1]$