



# Kolejka

Arezou i jej brat Borzou są bliźniętami. Otrzymali na urodziny wspianiałą zabawkę -- kolejkę. Zaczęli tworzyć sieć kolejową z  $n$  stacjami oraz  $m$  *jednokierunkowymi* torami. Stacje numerowane są kolejnymi liczbami naturalnymi od 0 do  $n - 1$  włącznie. Każdy tor rozpoczyna się na pewnej stacji i kończy na tej samej lub innej stacji. Każda stacja ma co najmniej jeden tor rozpoczynający się na niej.

Niektóre stacje są *stacjami ładującymi*. Ilekroć kolejka dociera do stacji ładującej, ładuje się do pełna. W pełni naładowana kolejka ma wystarczająco dużo energii, aby przejechać  $n$  kolejnych torów. To znaczy, że kolejka całkowicie się rozładuje dokładnie wtedy, gdy wjedzie na  $(n + 1)$ -szy tor od ostatniego ładowania.

Na każdej stacji znajduje się przełącznik, który może wskazywać na dowolny tor rozpoczynający się na danej stacji. Kolejka znajdująca się na danej stacji, opuszcza ją torem wskazywanym przez przełącznik na tej stacji.

Bliźniaki zamierzają zagrać w grę z użyciem ich kolejki. Podzielili już stacje pomiędzy sobą: każda stacja należy albo do Arezou albo do Borzou. Jest jedna kolejka. Na początku gry kolejka jest na stacji  $s$  i jest w pełni naładowana. Aby rozpocząć grę, właściciel stacji  $s$  ustawia przełącznik znajdujący się na stacji  $s$  na jeden z torów zaczynających się na tej stacji. Następnie kolejka jest włączana i zaczyna jechać po torach.

Ilekroć kolejka wjeżdża na stację po raz pierwszy, właściciel tej stacji ustawia przełącznik znajdujący się na niej. Gdy przełącznik zostanie ustawiony, pozostaje w tej pozycji do końca gry. A zatem, jeśli kolejka powróci na stację, na której już była wcześniej, opuści tę stację tym samym torem co poprzednio.

Ponieważ liczba stacji jest skończona, kolejka kiedyś zacznie jechać po *cyklu*. Cykl to ciąg *różnych* stacji  $c[0], c[1], \dots, c[k - 1]$  taki że kolejka opuszcza stację  $c[i]$  (dla  $0 \leq i < k - 1$ ) torem prowadzącym do stacji  $c[i + 1]$  oraz opuszcza stację  $c[k - 1]$  torem prowadzącym do stacji  $c[0]$ . Zauważ, że cykl może zawierać pojedynczą stację (tzn.  $k = 1$ ), jeśli kolejka opuszcza stację  $c[0]$  torem prowadzącym z powrotem do  $c[0]$ .

Arezou wygrywa grę, jeśli kolejka będzie jechać w nieskończoność, zaś Borzou wygrywa, gdy kolejka się rozładuje. Innymi słowy, jeśli istnieje co najmniej jedna stacja ładująca wśród  $c[0], c[1], \dots, c[k - 1]$ , kolejka może się ładować i jeździć po cyklu w nieskończoność -- wtedy Arezou wygrywa. W przeciwnym przypadku, kolejka się rozładuje (być może po kilkukrotnym przejechaniu cyklu) i wtedy wygrywa Borzou.

Dany jest opis sieci kolejowej. Arezou i Borzou mają zamiar zagrać  $n$  gier. W  $s$ -tej grze, dla

$0 \leq s \leq n - 1$ , kolejka będzie rozpoczynać bieg na stacji  $s$ . Twoim zadaniem jest ustalić, dla każdej gry, czy istnieje strategia gwarantująca zwycięstwo Arezou, niezależnie od ruchów Borzou.

## Szczegóły implementacji

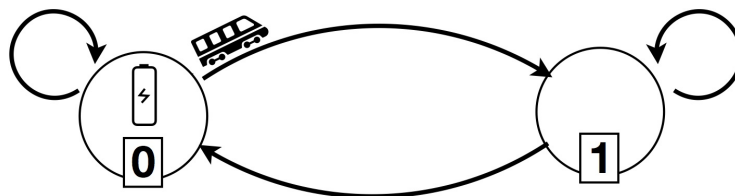
Należy zaimplementować następującą funkcję:

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

- $a$ : tablica długości  $n$ . Jeśli Arezou jest właścicielką stacji  $i$ ,  $a[i] = 1$ . W przeciwnym przypadku stacja należy do Borzou i  $a[i] = 0$ .
- $r$ : tablica długości  $n$ . Jeśli stacja  $i$  jest stacją ładującą,  $r[i] = 1$ . W przeciwnym przypadku,  $r[i] = 0$ .
- $u$  i  $v$ : tablice o długości  $m$ . Dla każdego  $0 \leq i \leq m - 1$ , istnieje jednokierunkowy tor rozpoczynający się przy stacji  $u[i]$  i kończący na stacji  $v[i]$ .
- Funkcja powinna zwrócić tablicę  $w$  długości  $n$ . Dla każdego  $0 \leq i \leq n - 1$ , wartość  $w[i]$  powinna być 1, jeśli Arezou może wygrać grę, która rozpoczyna się na stacji  $i$ , niezależnie od gry Borzou. W przeciwnym przypadku wartość  $w[i]$  powinna wynosić 0.

## Przykład

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



- Są 2 stacje. Borzou jest właścicielem stacji 0, która jest stacją ładującą. Arezou jest właścicielką stacji 1, która nie jest stacją ładującą.
- Są 4 tory  $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$  oraz  $(1, 1)$ , gdzie  $(i, j)$  oznacza jednokierunkowy tor ze stacji  $i$  do stacji  $j$ .
- Rozważmy grę, w której kolejka jest umieszczona początkowo na stacji 0. Jeśli Borzou ustawi przełącznik na stacji 0 w stronę toru  $(0, 0)$ , kolejka będzie jechać w nieskończoność po cyklu wzdłuż tego toru (zauważ, że stacja 0 jest stacją ładującą). W tym przypadku, Arezou wygrywa. W przeciwnym przypadku, jeśli Borzou ustawi przełącznik na stacji 0 w stronę toru  $(0, 1)$ , Arezou może ustawić przełącznik na stacji 1 w stronę toru  $(1, 0)$ . Jeśli tak zrobi, kolejka będzie jechać w nieskończoność po cyklu złożonym z obu stacji. W tym przypadku Arezou także wygrywa, ponieważ stacja 0 jest stacją ładującą, więc pociąg się nie zatrzyma. Ostatecznie, Arezou może wygrać grę, niezależnie od ruchów Borzou.
- Analogicznie, w grze rozpoczynającej się na stacji 1 Arezou także może wygrać, niezależnie od ruchów Borzou. Funkcja powinna zwrócić  $[1, 1]$ .

## Ograniczenia

- $1 \leq n \leq 5000$ .
- $n \leq m \leq 20\,000$ .
- Jest co najmniej jedna stacja ładująca.
- Jest co najmniej jeden tor rozpoczynający się na każdej stacji.
- Tory mogą rozpoczynając się i kończyć na tej samej stacji (tzn.  $u[i] = v[i]$ ).
- Tory są parami różne. Innymi słowy, nie ma dwóch takich  $i, j$  ( $0 \leq i < j \leq m - 1$ ), że  $u[i] = u[j]$  i  $v[i] = v[j]$ .
- $0 \leq u[i], v[i] \leq n - 1$  (dla każdego  $0 \leq i \leq m - 1$ ).

## Podzadania

1. (5 punktów) Dla każdego  $0 \leq i \leq m - 1$ , albo  $v[i] = u[i]$  albo  $v[i] = u[i] + 1$ .
2. (10 punktów)  $n \leq 15$ .
3. (11 punktów) Wszystkie stacje należą do Arezou.
4. (11 punktów) Wszystkie stacje należą do Borzou.
5. (12 punktów) Jest dokładnie jedna stacja ładująca.
6. (51 punktów) Brak dodatkowych ograniczeń.

## Przykładowa sprawdzaczka

Przykładowa sprawdzaczka odczytuje wejście w następującym formacie:

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

Przykładowa sprawdzaczka wypisuje zwróconą wartość funkcji `who_wins` w następującym formacie:

- wiersz 1:  $w[0] \ w[1] \ \dots \ w[n - 1]$ .