



## IOI 2017 Translation

NLD

## Toy Train

Arezou en haar broer Borzou zijn een tweeling. Ze hebben voor hun verjaardig een speelgoedtreinset gekregen waarmee ze een spoornetwerk hebben gebouwd van  $n$  stations en  $m$  eenrichtingssporen. De stations zijn genummerd van 0 tot en met  $n - 1$ . Elk spoor begint in een station en eindigt in hetzelfde of een ander station. Vanuit elk station vertrekt tenminste één spoor.

Sommige stations zijn *oplaadstations*. Wanneer een trein aankomt in een oplaadstation wordt hij volledig opgeladen. Een volledige opgeladen trein heeft voldoende energie om over  $n$  opeenvolgende sporen te rijden. Wanneer een trein het  $(n + 1)$ -de spoor oprijdt na de laatste keer opgeladen te zijn, komt de trein tot stilstand.

Op elk station is er een wissel die naar een uitgaand spoor van dat station wijst. Als een trein op dit station is, dan verlaat het dit station via het spoor waar de wissel op dat station naar wijst.

De tweeling gaat een spel spelen met hun treinset. Ze hebben alle stations onder elkaar verdeeld: elk station is ofwel van Arezou ofwel van Borzou. Er is maar één trein. In het begin van het spel staat de trein in station  $s$  en is hij volledig opgeladen. Om het spel te starten, wijst de eigenaar van station  $s$  de wissel van station  $s$  naar een van de uitgaande sporen. Vervolgens zetten ze de trein aan en begint deze over de sporen te rijden.

Telkens als de trein voor de eerste keer een station binnenrijdt bepaalt de eigenaar van dat station naar welk uitgaand spoor de wissel wijst. Een ingestelde wissel blijft voor de rest van het spel in dezelfde positie staan. Dus als een trein een station opnieuw binnenkomt zal het dit station verlaten over hetzelfde spoor als voorheen.

Aangezien er een eindig aantal stations is, zal de trein uiteindelijk in een *lus* terechtkomen. Een lus is een opeenvolging van *verschillende* stations  $c[0], c[1], \dots, c[k - 1]$  zodanig dat de trein uit station  $c[i]$  (met  $0 \leq i < k - 1$ ) vertrekt over het spoor naar station  $c[i + 1]$  en hij vanuit station  $c[k - 1]$  over het spoor naar station  $c[0]$  doorgaat. Let op dat een lus mogelijk uit slechts uit één station bestaat (met  $k = 1$ ) als er een spoor vertrekt uit station  $c[0]$  over een spoor dat weer naar  $c[0]$  gaat.

Arezou wint het spel als de train eindeloos blijft doorgaan. Borzou wint als de trein zonder energie stil komt te staan. Anders gezegd, als er minstens één oplaadstation is onder de  $c[0], c[1], \dots, c[k - 1]$  kan de trein eindeloos opladen en rondrijden en wint Arezou. Als de energie (eventueel na een aantal lussen gemaakt te hebben) opraakt dan wint Borzou.

Je krijgt de beschrijving van het spoornetwerk. Arezou en Borzou gaan  $n$  spelletjes spelen. In het  $s$ -de spel, voor  $0 \leq s \leq n - 1$ , begint de trein in station  $s$ . Jouw taak is om, voor elk spelletje, te bepalen of er een strategie bestaat voor Arezou die garandeert dat ze wint, onafhankelijk van hoe Borzou speelt.

## Implementatiedetails

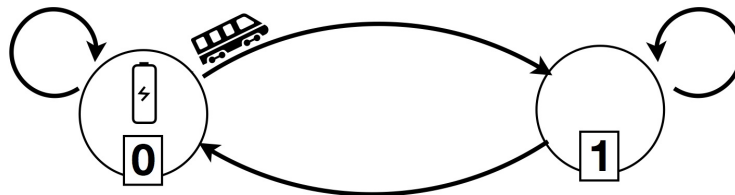
Implementeer de volgende functie:

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

- $a$ : een array van lengte  $n$ . Als station  $i$  van Arezou is geldt  $a[i] = 1$ . Anders is het station van Borzou en geldt  $a[i] = 0$ .
- $r$ : een array van lengte  $n$ . Als station  $i$  een oplaadstation is geldt  $r[i] = 1$ , anders geldt  $r[i] = 0$ .
- $u$  en  $v$  zijn arrays van lengte  $m$ . Voor alle  $0 \leq i \leq m - 1$  is er een eenrichtingsspoor vertrekkend vanuit station  $u[i]$  en dat eindigt in station  $v[i]$ .
- De functie moet een array  $w$  van lengte  $n$  retourneren. Voor alle  $0 \leq i \leq n - 1$  moet de waarde van  $w[i]$  gelijk aan 1 zijn als Arezou het spelletje dat begint in station  $i$  kan winnen, onafhankelijk van hoe Borzou speelt. Anders moet de waarde van  $w[i]$  gelijk aan 0 zijn.

## Voorbeeld

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



- Er zijn 2 stations. Station 0 is van Borzou en is een oplaadstation. Station 1 is van Arezou en is geen oplaadstation.
- Er zijn 4 sporen  $(0,0)$ ,  $(0,1)$ ,  $(1,0)$  en  $(1,1)$ , waarbij  $(i,j)$  aangeeft dat er een eenrichtingsspoor bestaat van station  $i$  naar station  $j$ .
- Beschouw het spel waarbij de trein begint in station 0. Als Borzou de wissel van station 0 naar het spoor  $(0,0)$  wijst, rijdt de trein eindelijk in een lus langs dit spoor (merk op dat station 0 een oplaadstation is). In dit geval wint Arezou. Als Borzou de wissel van station 0 naar spoor  $(0,1)$  wijst, kan Arezou de wissel van station 1 naar  $(1,0)$  wijzen. Als dit gebeurt zal de trein eindelijk door beide stations rijden. Arezou wint opnieuw, omdat station 0 een oplaadstation is en de trein nooit zal stoppen. Daarom kan Arezou het spelletje winnen, onafhankelijk van wat Borzou doet.
- Met een analoge redenering kan Arezou ook winnen in het spelletje waarbij de trein in station

1 vertrekt, onafhankelijk hoe Borzou speelt. De functie moet dus  $[1, 1]$  teruggeven.

## Randvoorwaarden

- $1 \leq n \leq 5000$ .
- $n \leq m \leq 20\,000$ .
- Er is tenminste één laadstation.
- Er vertrekt tenminste één spoor vanuit elk station.
- Er kunnen sporen zijn die in hetzelfde station aankomen als waar ze vertrekken (dus,  $u[i] = v[i]$ ).
- Elk spoor is uniek. Met andere woorden zijn er geen indices  $i$  en  $j$  ( $0 \leq i < j \leq m - 1$ ) zodat  $u[i] = u[j]$  en  $v[i] = v[j]$ .
- $0 \leq u[i], v[i] \leq n - 1$  (voor alle  $0 \leq i \leq m - 1$ ).

## Subtaken

1. (5 punten) Voor alle  $0 \leq i \leq m - 1$ , ofwel  $v[i] = u[i]$  of  $v[i] = u[i] + 1$ .
2. (10 punten)  $n \leq 15$ .
3. (11 punten) Alle stations zijn van Arezou
4. (11 punten) Alle stations zijn van Borzou.
5. (12 punten) Er is exact één oplaadstation.
6. (51 punten) Geen bijkomende beperkingen.

## Voorbeeldgrader

De voorbeeldgrader leest de input in het volgende formaat:

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

De voorbeeldgrader print de terugkeerwaarde van `who_wins` in het volgende formaat:

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