



BRIO™-tåg

Arezou och hennes bror Borzou är tvillingar. De har båda fått ett otroligt BRIO™-tågset på deras födelsedag, och använde det för att bygga ett järnvägssystem med n stationer och m enkelriktade spår. Stationerna är numrerade från 0 till $n - 1$. Varje spår börjar på en station och slutar på någon station (möjligtvis samma). Det finns minst ett spår som börjar på varje station.

Vissa stationer är laddningsstationer. När ett tåg ankommer en laddningsstation blir det fulladdat. Ett fulladdat tåg har precis tillräckligt med energi för att färdas längs n på varandra följande spår. D.v.s., tåget får slut på energi precis när det ska åka in på det $(n + 1)$:a spåret efter sin senaste laddning.

På varje station finns det en växel som kan användas för att välja ett spår som börjar på stationen. När ett tåg lämnar en viss station gör den det genom att åka på det av växeln valda spåret.

Tvillingarna ska nu spela ett spel med sitt tåg. De har redan delat upp samtliga stationer mellan sig: varje station ägs av antingen Arezou eller Borzou. Det finns ett enda tåg. I början av spelet är tåget på station s och fulladdat. För att börja spelet väljer ägaren av station s vilket spår växeln på station s ska peka på bland de som börjar på station s . Sedan sätter de på tåget varpå det börjar färdas längs spåren.

När ett tåg för första gången ankommer en station väljer stationens ägare vilket av de spår som börjar på stationen som stationens växel ska peka på. När en växel har pekats ändras inte dess position något mer under spelets gång. Detta betyder att om ett tåg ankommer en station det tidigare besökt kommer det lämna stationen på samma spår som tidigare.

Eftersom det finns ett ändligt antal stationer kommer tåget eventuellt börja resa längs en cykel. En cykel är en sekvens av parvis olika stationer $c[0], c[1], \dots, c[k - 1]$ sådan att tåget lämnar station $c[i]$ (för $0 \leq i < k - 1$) med ett spår som går till station $c[i + 1]$ och lämnar station $c[k - 1]$ med ett spår som går till station $c[0]$. Notera att en cykel kan bestå av en enda station (d.v.s., den har $k = 1$) om tåget lämnar station $c[0]$ med ett spår som går tillbaka till $c[0]$.

Arezou vinner spelet om tåget fortsätter åka i all oändlighet, och Borzou vinner om tåget får slut på energi. Med andra ord, om det finns minst en laddningsstation bland $c[0], c[1], \dots, c[k - 1]$ kan tåget ladda om och åka längs cykeln i all oändlighet, och Arezou vinner. Annars kommer det få slut på energi (möjligtvis efter flera färder längs cykeln), och Borzou vinner.

Du får en beskrivning över järnvägssystemet. Arezou och Borzou ska spela n spel. I det s :te spelet, för $0 \leq s \leq n - 1$ är tåget ursprungligen på station s . Din uppgift är att, för varje spel, avgöra om det finns en strategi som Arezou kan använda sig av för att garantera att hon vinner, oavsett hur Borzou spelar.

Intrikat implementationsinformation

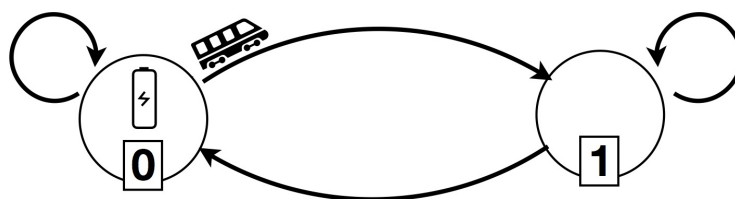
Du ska implementera följande procedur:

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

- a : array av längd n . Om Arezou äger station i är $a[i] = 1$. Annars äger Borzou station i och $a[i] = 0$.
- r : array av längd n . Om station i är en laddningsstation är $r[i] = 1$. Annars är $r[i] = 0$.
- u och v : arrayer av längd m . För alla $0 \leq i \leq m - 1$, det finns ett enkelriktat spår som börjar på station $u[i]$ och slutar på station $v[i]$.
- Proceduren ska returnera en array w av längd n . För alla $0 \leq i \leq n - 1$ ska $w[i]$ vara 1 om Arezou kan vinna spelet där tåget börjar på station i , oavsett hur Borzou spelar. Annars ska $w[i]$ vara 0.

Exemplexekvering

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



- Det finns 2 stationer. Borzou äger station 0, som är en laddningsstation. Arezou äger station 1, som inte är en laddningsstation.
- Det finns 4 spår $(0, 0)$, $(0, 1)$, $(1, 0)$, och $(1, 1)$, där (i, j) betecknar ett enkelriktat spår från station i till station j .
- Betrakta spelet där tåget börjar på station 0. Borzou pekar växeln på station 0 mot spåret $(0, 0)$, så tåget kommer i all oändlighet att färdas längs detta spår (notera att station 0 är en laddningsstation). I detta fall vinner Arezou. Om Borzou istället pekar växeln på station 0 mot spåret $(0, 1)$, kan Arezou peka växeln på station 1 mot $(1, 0)$. Då kommer tåget cykla genom båda stationer oändligt länge. Återigen vinner Arezou, eftersom station 0 är en laddningsstation. Således kan Arezou vinna spelet, oavsett vad Borzou gör.
- Med ett liknande resnomenag kan man visa att om tåget börjar på station 1 kommer Arezou också vinna oavsett hur Borzou spelar. Proceduren ska alltså returnera $[1, 1]$.

• Gränser

- $1 \leq n \leq 5000$.
- $n \leq m \leq 20\,000$.
- Det finns minst en laddningsstation.
- Det finns minst ett spår som börjar på varje station.
- Det kan finnas spår som börjar och slutar på samma station (d.v.s., $u[i] = v[i]$).
- Alla spår är olika. Med andra ord finns inga index i and j ($0 \leq i < j \leq m - 1$) så att både $u[i] = u[j]$ och $v[i] = v[j]$ gäller.
- $0 \leq u[i], v[i] \leq n - 1$ (för alla $0 \leq i \leq m - 1$).

Delproblem

1. (5 poäng) För alla $0 \leq i \leq m - 1$, gäller antingen $v[i] = u[i]$ eller $v[i] = u[i] + 1$.
2. (10 poäng) $n \leq 15$.
3. (11 poäng) Arezou äger alla stationer.
4. (11 poäng) Borzou äger alla stationer.
5. (12 poäng) Det finns exakt en laddningsstation.
6. (51 poäng) Inga ytterligare gränser.

Exempeldomare

Exempeldomaren läser indata på följande format:

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

Exempeldomaren skriver ut returvärdet från `who_wins` på följande format:

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