



## Žaislinis traukinukas

Arezou ir jos brolis Borzou yra dvyniai. Gimtadienio proga jie gavo traukinuko rinkinį. Jie sukonstravo bėgių ir stočių sistemą iš  $n$  stočių ir  $m$  vienos krypties bėgių. Stotys sunumeruotos nuo 0 iki  $n - 1$ . Kiekvieni bėgiai prasideda vienoje stotyje ir baigiasi toje pačioje arba kitoje stotyje. Iš kiekvienos stoties išeina bent vieneri bėgiai.

Kai kuriose stotys yra *pakrovimo stotys*. Jeigu traukinukas atvažiuoja į tokią stotį, jis pilnai pakraunamas. Pilnai pakrautam traukinukui užtenka energijos pervažiuoti  $n$  vienas po kito einančių bėgių. Tai yra traukinukas pilnai išsikrauna, kai privažiuoja  $(n + 1)$ -us bėgius po paskutinio pakrovimo.

Kiekvienoje stotyje yra perjungiklis, kurį galima nukreipti į bet kuriuos iš tos stoties išeinančius bėgius. Kai traukinukas išvažiuoja stoties, jis važiuos tai bėgiais, į kuriuos jį nukreipia perjungiklis.

Dvyniai vienu traukinuku žaidžia tokį žaidimą. Jie tarpusavyje pasidalino stotis: kiekvienos stoties savininkas yra arba Arezou, arba Borzou. Pradiniu momentu traukinukas yra pilnai pakrautas ir stovi stotyje  $s$ . Žaidimą pradeda stoties  $s$  savininkas, kuris nukreipia stotyje  $s$  esantį perjungiklį į pasirinktus bėgius, išeinančius iš stoties  $s$ . Tuomet paleidžiamas traukinukas ir jis ima važiuoti bėgiais.

Kai tik traukinukas pirmą kartą pasiekia kurią nors stotį, tos stoties savininkas nukreipia perjungiklį į pasirinktus bėgius. Tai padarius, daugiau žaidime to perjungiklio pozicija nekeičiama. Taigi, jei traukinukas antrą kartą atvažiuoja į tą pačią stotį, jis išvažiuos tai pačiais bėgiais.

Kadangi stočių skaičius baigtinis, anksčiau ar vėliau traukinukas ims važiuoti ratu (*ciklu*). Ciklas yra tokia skirtingų stočių seka  $c[0], c[1], \dots, c[k - 1]$ , kad traukinys važiuoja iš stoties  $c[i]$  ( $0 \leq i < k - 1$ ) bėgiais, vedančiais į stotį  $c[i + 1]$ , o iš stoties  $c[k - 1]$  važiuoja bėgiais, vedančiais į stotį  $c[0]$ . Atkreipkite dėmesį, kad ciklą gali sudaryti vienintelė stotis (t.y. kai  $k = 1$ ) jei traukinys važiuoja iš stoties  $c[0]$  bėgiais, vedančiais atgal į  $c[0]$ .

Arezou laimi žaidimą, jei traukinukas važinėja ciklu be sustojimo, o Borzou laimi, jei traukinukas išsikrauna ir sustoja. Kitaip sakant, jei tarp stočių  $c[0], c[1], \dots, c[k - 1]$  yra bent viena pakrovimo stotis, jis visą laiką važinės ratu, ir Arezou laimės. Priešingu atveju jis išsikraus (galbūt pravažiavęs ciklą keletą kartų), sustos ir Borzou laimės.

Jums duota žaislinio geležinkelio schema. Arezou ir Borzou ruošiasi žaisti  $n$  žaidimų.  $s$ -ajame žaidime ( $0 \leq s \leq n - 1$ ) pradiniu momentu traukinukas bus stotyje  $s$ . Kiekvienam žaidimui nustatykite, ar egzistuoja žaidimo strategija, kuria pasinaudojusi Arezou laimėtų, nesvarbu, kaip žaistų Borzou.

## Realizacija

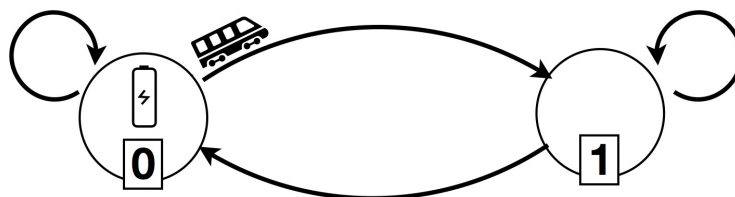
Parašykite šią procedūrą:

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

- $a$ :  $n$  ilgio masyvas. Jei  $i$ -osios stoties savininkė yra Arezou,  $a[i] = 1$ . Jei ne,  $i$ -osios stoties savininkas yra Borzou ir  $a[i] = 0$ .
- $r$ :  $n$  ilgio masyvas. Jei  $i$ -oji stotis yra pakrovimo stotis,  $r[i] = 1$ . Jei ne,  $r[i] = 0$ .
- $u$  ir  $v$ :  $m$  ilgio masyvai. Kiekvienam  $i$  ( $0 \leq i \leq m - 1$ ) yra vienos krypties bėgiai, prasidedantys stotyje  $u[i]$  ir pasibaigiantys stotyje  $v[i]$ .
- Ši procedūra turi grąžinti  $n$  ilgio masyvą  $w$ . Kiekvienam  $i$  ( $0 \leq i \leq n - 1$ ),  $w[i]$  turi būti 1, jei Arezou gali laimėti žaidimą, pradėtą žaisti  $i$ -ojoje stotyje, nesvarbu, kaip žaistų Borzou. Kitu atveju  $w[i]$  turi būti 0.

## Pavyzdys

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



- Yra 2 stotys. Borzou yra stoties 0 savininkas ir ši stotis yra pakrovimo stotis. Arezou yra stoties 1 savininkė ir ši stotis nėra pakrovimo stotis.
- Yra 4 bėgiai  $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$ , ir  $(1, 1)$ , kur  $(i, j)$  žymi vienos krypties bėgius iš stoties  $i$  į stotį  $j$ .
- Panagrinėkime žaidimą, kurio pradžioje traukinys stovi stotyje 0. Jei Borzou nukreipia perjungiklį stotyje 0 link bėgių  $(0, 0)$ , traukinys visą laiką važiuos ratu šiais bėgiais. Kadangi stotis 0 yra pakrovimo stotis, Arezou laimi. Kitu atveju, jei Borzou nukreipia perjungiklį stotyje 0 link bėgių  $(0, 1)$ , Arezou gali nukreipti perjungiklį stotyje 1 link  $(1, 0)$ . Jei tai nutiks, traukinukas nesustodamas ratu važinės tarp abiejų stočių. Vėl laimės Arezou, nes stotis 0 yra pakrovimo stotis, taigi traukinukas neišsikraus. Taigi, Arezou laimės, nesvarbu, ką darytų Borzou.
- Panašiai samprotaudami gausime, kad jei žaidimas prasidės stotyje 1, Arezou laimės, nesvarbu, kaip žaistų Borzou. Todėl procedūra turi grąžinti  $[1, 1]$ .

## Ribojimai

- $1 \leq n \leq 5000$ .
- $n \leq m \leq 20\,000$ .
- Yra bent viena pakrovimo stotis.

- Iš kiekvienos stoties išeina bent vieneri bėgiai.
- Bėgiai gali prasidėti ir pasibaigti toje pačioje stotyje (t.y.,  $u[i] = v[i]$ ).
- Visi bėgiai yra skirtingi, kitaip sakant, nėra tokių dviejų indeksų  $i$  ir  $j$  ( $0 \leq i < j \leq m - 1$ ), kur  $u[i] = u[j]$  ir  $v[i] = v[j]$ .
- $0 \leq u[i], v[i] \leq n - 1$  (visiems  $0 \leq i \leq m - 1$ ).

## Dalinės užduotys

1. (5 taškai) Visiems  $0 \leq i \leq m - 1$ , galioja  $v[i] = u[i]$  arba  $v[i] = u[i] + 1$ .
2. (10 taškų)  $n \leq 15$ .
3. (11 taškų) Visos stotys priklauso Arezou.
4. (11 taškų) Visos stotys priklauso Borzou.
5. (12 taškų) Yra lygiai viena pakrovimo stotis.
6. (51 taškas) Papildomų ribojimų nėra.

## Pavyzdinis vertintojas

Pavyzdinis vertintojas skaito pradinis duomenis tokiu formatu:

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

Pavyzdinis vertintojas gražina `who_wins` reikšmę tokiu formatu:

- 1-oji eilutė:  $w[0] \ w[1] \ \dots \ w[n - 1]$