



## Oyuncak Tren (Toy Train)

Arezou ve erkek kardeşi Borzou ikizdir. Doğum günlerinde hediye olarak harika bir oyuncak tren almışlardır. Kardeşler, bu oyuncaklarıyla  $n$  istasyonu ve  $m$  *tek-yönlü* rayı olan bir tren sistemi kurdular. İstasyonlar 0 'dan  $n - 1$  'e kadar numaralandırılmıştır. Her bir ray, bir istasyondan çıkar ve aynı istasyona veya farklı bir istasyona gider. Her bir istasyondan çıkan en az bir ray vardır.

Bazı istasyonlar *şarj istasyonlarıdır*. Bir tren bir şarj istasyonuna geldiğinde, tam olarak şarj olur. Tam şarjlı bir tren  $n$  tane ardışık rayı dolaşacak kadar enerjiye sahiptir. Yani, tren son şarjdan sonra,  $(n + 1)$ 'inci raya girerken enerjisi biter ve durur.

Her bir istasyonda, o istasyondan ayrılan her bir rayı gösterebilen bir anahtar bulunmaktadır. Tren bir istasyondayken, o istasyondaki anahtarın gösterdiği rayı kullanarak ayrılır.

İkizler trenleri ile bir oyun oynayacaktır. Aralarında tüm istasyonları paylaşmışlardır: Her bir istasyonu ya Arezou ya da Borzou sahiplenmiştir. Tek bir tren vardır. Oyunun başlangıcında tren  $s$  istasyonundadır ve tam olarak şarjlıdır. Oyunu başlatmak için,  $s$  istasyonunun sahibi  $s$  istasyonundaki anahtarı, istasyondan giden raylardan birini gösterecek şekilde ayarlar. Daha sonra treni çalıştırır ve tren raylar üzerinden hareket etmeye başlar.

Tren bir istasyona ilk defa geldiğinde, o istasyonun sahibi anahtarı ayarlar. Anahtar ayarlandığında, oyunun geri kalanı için anahtar aynı yönde aynı pozisyonda kalır. Dolayısıyla, bir tren daha önce ziyaret ettiği bir istasyona tekrar gelirse, daha önce kullandığı ray ile istasyonu terk edecektir.

Sonlu sayıda istasyon olduğundan, tren eninde sonunda bir *döngü* içinde olacaktır. Döngü, *farklı* istasyon sırasıdır  $c[0], c[1], \dots, c[k - 1]$  öyle ki tren  $c[i]$  ( $0 \leq i < k - 1$  için) istasyonundan  $c[i + 1]$  istasyonuna giden rayı kullanarak ayrılır, ve  $c[k - 1]$  istasyonundan  $c[0]$  istasyonuna giden rayı kullanarak ayrılır. Not: Bir döngü tek bir istasyon içerebilir (yani  $k = 1$  olur), eğer tren  $c[0]$  istasyonundan tekrar  $c[0]$  istasyonuna giden rayı kullanırsa.

Eğer tren sonsuza kadar yolunda devam ederse Arezou oyunu kazanır ve eğer trenin enerjisi biterse Borzou kazanır. Başka bir deyişle,  $c[0], c[1], \dots, c[k - 1]$  içinde en az bir şarj istasyonu varsa tren tekrar şarj edilebilir ve sonsuza kadar döngüye girer, dolayısıyla Arezou kazanır. Diğer durumda, enerjisi biter (muhtemelen bir döngüye bir kaç kez girdikten sonra) ve Borzou kazanır.

Ray sisteminin tanımı size veriliyor. Arezou ve Borzou  $n$  tane oyun oynayacaktır.  $s$ . oyunda,  $0 \leq s \leq n - 1$  için, tren başlangıçta  $s$  istasyonunda olacaktır. Göreviniz, her bir oyun için, Borzou'nun nasıl oynadığından bağımsız olmak üzere, Arezou'nun kazanmasını garanti eden bir strateji olup olmadığını bulmanızdır.

## Gerçekleştirim detayları

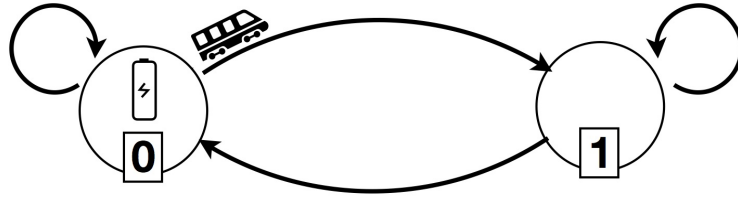
Aşağıdaki prosedürü kodlamalısınız:

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

- $a$ :  $n$  uzunluğunda dizi. Eğer Arezou  $i$  istasyonuna sahipse,  $a[i] = 1$ . Diğer durumda, Borzou  $i$  istasyonuna sahiptir ve  $a[i] = 0$ .
- $r$ :  $n$  uzunluğunda dizi. Eğer  $i$  istasyonu sarj istasyonu ise,  $r[i] = 1$ . Diğer durumda  $r[i] = 0$ .
- $u$  ve  $v$ :  $m$  uzunluğunda diziler. Tüm  $0 \leq i \leq m - 1$  için,  $u[i]$  istasyonundan ayrılan ve  $v[i]$  istasyonuna gelen tek-yönlü bir ray vardır.
- Bu prosedür  $n$  uzunluğunda  $w$  dizisi dönmelidir. Tüm  $0 \leq i \leq n - 1$  için, eğer Arezou, Borzou'nun nasıl oynadığından bağımsız olmak üzere oyunu kazanabilirse,  $w[i]$  değeri 1 olmalıdır. Aksi takdirde,  $w[i]$  değeri 0 olmalıdır.

## Örnek

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



- 2 istasyon vardır. Borzou 0 istasyonunun sahibidir ve bu istasyon sarj istasyonudur. Arezou 1 istasyonunun sahibidir ve bu istasyon şarj istasyonu değildir.
- 4 tane ray vardır  $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$ , ve  $(1, 1)$ , burada  $(i, j)$ ,  $i$  istasyonundan  $j$  istasyonuna giden tek-yönlü bir rayı belirtir.
- Trenin başlangıçta 0 istasyonunda olduğu oyunu göz önüne alın. Eğer Borzou 0 istasyonundaki anahtarı  $(0, 0)$  rayı yönünde çevirirse, tren sonsuza kadar bu ray üzerinde döngüye girer (not: 0 istasyonu sarj istasyonudur). Bu durumda Arezou kazanır. Diğer durumda, Borzou 0 istasyonundaki anahtarı  $(0, 1)$  rayı yönünde çevirirse Arezou 1 istasyonundaki anahtarı  $(1, 0)$  yönüne çevirebilir. Eğer bu olursa, tren sonsuza kadar bu iki istasyon üzerinde döngüye girer. Yine, Arezou kazanır, çünkü 0 istasyonu sarj istasyonudur ve tren durmayacaktır. Dolayısıyla, Borzou'nun yaptığı hamlelerden bağımsız olarak Arezou oyunu kazanabilir.
- Benzer bir mantıkla, 1 istasyonundan başlayan oyunda da Borzou'nun nasıl oynadığından bağımsız olacak şekilde Arezou oyunu kazanabilir. Yani, prosedür  $[1, 1]$  dönmelidir.

## Kısıtlar

- $1 \leq n \leq 5000$ .
- $n \leq m \leq 20\,000$ .
- En az bir sarj istasyonu vardır.

- Her bir istasyondan çıkan en az bir ray vardır.
- Aynı istasyondan çıkan ve aynı istasyona gelen raylar olabilir (yani  $u[i] = v[i]$ ).
- Her bir ray farklıdır. Başka bir deyişle,  $u[i] = u[j]$  ve  $v[i] = v[j]$  olacak şekilde iki  $i$  ve  $j$  indisi ( $0 \leq i < j \leq m - 1$ ) yoktur.
- $0 \leq u[i], v[i] \leq n - 1$  (tüm  $0 \leq i \leq m - 1$  için).

## Altgörevler

1. (5 puan) Tüm  $0 \leq i \leq m - 1$  için, ya  $v[i] = u[i]$  ya da  $v[i] = u[i] + 1$ .
2. (10 puan)  $n \leq 15$ .
3. (11 puan) Arezou tüm istasyonlara sahiptir.
4. (11 puan) Borzou tüm istasyonlara sahiptir.
5. (12 puan) Tam olarak bir tane şarj istasyonu vardır.
6. (51 puan) Ek kısıt yoktur.

## Örnek değerlendirici

Örnek değerlendirici girdiyi aşağıdaki formatta okur:

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

Örnek değerlendirici `who_wins` in geri dönme değerini aşağıdaki formatta yazar:

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