



Toy Train

Arezou e o seu irmão são gêmeos. Eles receberam um incrível conjunto de comboios de brincar para o seu aniversário e usaram-no para construir um sistema de caminhos de ferro com n estações e m linhas *unidirecionais*. As estações estão numeradas de 0 a $n - 1$. Cada linha começa numa estação e termina na mesma ou numa estação diferente. Há pelo menos uma linha que começa em cada estação.

Algumas estações são *estações de carregamento*. Quando um comboio chega a uma estação de carregamento, ele fica completamente carregado. Um comboio completamente carregado tem bateria suficiente para se movimentar por n linhas consecutivas. Isto é, um comboio fica sem bateria quando entra na $(n + 1)$ -ésima linha após ter sido carregado.

Em cada estação existe um interruptor que pode ser apontado para qualquer uma das linhas que começa nessa estação. Quando um comboio está numa estação, ele sai usando a linha para a qual o interruptor dessa estação aponta.

Os gêmeos estão a jogar um jogo com o seu comboio. Eles já dividiram todas as estações entre si: cada estação pertence ou à Arezou ou ao Borzou. Existe apenas um único comboio. No início do jogo o comboio está na estação s e está completamente carregado. Para iniciar o jogo, o dono da estação s aponta o interruptor na estação s para uma das linhas que começa na estação s . De seguida, eles ligam o comboio e o comboio começa a andar pelas linhas.

Sempre que um comboio entra numa estação pela primeira vez, o dono dessa estação escolhe a posição do interruptor nessa estação. Sempre que um interruptor é colocado numa posição, ele fica na mesma posição para o resto do jogo. Assim, se um comboio reentrar numa estação que visitou antes, ele irá sair pela mesma estação que saiu anteriormente.

Visto que há um número finito de estações, o comboio eventualmente entrará num *ciclo*. Um ciclo é uma sequência de estações *distintas* $c[0], c[1], \dots, c[k - 1]$ tal que o comboio sai da estação $c[i]$ (para $0 \leq i < k - 1$) pela linha que termina na estação $c[i + 1]$, e sai da estação $c[k - 1]$ usando a linha que termina na estação $c[0]$. Nota que um ciclo pode consistir de uma única estação (i.e. $k = 1$) se o comboio sair da estação $c[0]$ usando uma linha que volta à estação $c[0]$.

Arezou ganha o jogo se o comboio continua a andar indefinidamente, e o Borzou ganha se o comboio ficar sem bateria. Por outras palavras, se existir pelo menos uma estação de carregamento nas $c[0], c[1], \dots, c[k - 1]$ então o comboio pode recarregar e ficar em ciclo indefinidamente, e a Arezou ganha. Caso contrário, ficará sem bateria (possivelmente após percorrer o ciclo várias vezes), e o Borzou ganha.

Será dada a descrição do sistema de caminhos de ferro. A Arezou e o Borzou jogarão n jogos. No

s -ésimo jogo, para $0 \leq s \leq n - 1$, o comboio estará inicialmente na estação s . A tua tarefa é encontrar, para cada jogo, se existe uma estratégia para a Arezou que garanta que ela ganhe, independentemente de como o Borzou jogar.

Detalhes de implementação

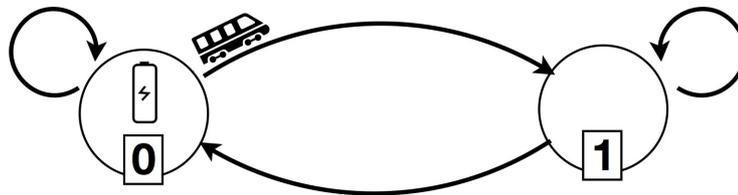
Deves implementar a seguinte função:

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

- a : vetor de comprimento n . Se a Arezou é dona da estação i , $a[i] = 1$. Caso contrário, o Borzou é dono da estação i e $a[i] = 0$.
- r : vetor de comprimento n . Se a estação i é uma estação de carregamento, $r[i] = 1$. Caso contrário, $r[i] = 0$.
- u e v : vetores de comprimento m . Para todo $0 \leq i \leq m - 1$, existe uma linha unidirecional que começa na estação $u[i]$ e acaba na estação $v[i]$.
- Esta função deve retornar um vetor w de comprimento n . Para cada $0 \leq i \leq n - 1$, o valor de $w[i]$ deve ser 1 se a Arezou consegue ganhar o jogo que começa na estação i , independentemente de como o Borzou jogar. Caso contrário, o valor de $w[i]$ deverá ser 0.

Exemplo

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



- Existem 2 estações. O Borzou é o dono da estação 0 que é uma estação de carregamento. A Arezou é a dona da estação 1 que não é uma estação de carregamento.
- Existem 4 linhas $(0, 0)$, $(0, 1)$, $(1, 0)$, e $(1, 1)$, onde (i, j) denota uma linha unidirecional da estação i para a estação j .
- Considera o jogo em que o comboio está inicialmente colocado na estação 0. Se o Borzou colocar o interruptor da estação 0 na linha $(0, 0)$, o comboio irá ficar indefinidamente em ciclo por esta estação (nota que a estação 0 é uma estação de carregamento). Neste caso, a Arezou ganha. Caso contrário, se o Borzou colocar o interruptor na linha $(0, 1)$, a Arezou pode colocar o interruptor na estação 1 da linha $(1, 0)$. Se isto acontecer, o comboio estará indefinidamente em ciclo por ambas as estações. Novamente, a Arezou ganha, visto que a estação 0 é uma estação de carregamento e o comboio não irá parar. Assim, a Arezou ganhará o jogo, independentemente do que o Borzou fizer.
- Por um raciocínio similar, no jogo que começa na estação 1 a Arezou também consegue

ganhar, independentemente do que o Borzou fizer. Consequentemente, a função deve retornar $[1, 1]$.

Restrições

- $1 \leq n \leq 5000$.
- $n \leq m \leq 20\,000$.
- Existe pelo menos uma estação de carregamento.
- Existe pelo menos uma linha a começar em cada estação.
- Podem haver linhas que começam e terminam na mesma estação (i.e., $u[i] = v[i]$).
- Cada linha é distinta. Por outras palavras, não existem dois índices i e j ($0 \leq i < j \leq m - 1$) tais que $u[i] = u[j]$ e $v[i] = v[j]$.
- $0 \leq u[i], v[i] \leq n - 1$ (para todo $0 \leq i \leq m - 1$).

Subtarefas

1. (5 pontos) Para todo $0 \leq i \leq m - 1$, ou $v[i] = u[i]$ ou $v[i] = u[i] + 1$.
2. (10 pontos) $n \leq 15$.
3. (11 pontos) A Arezou é dona de todas as estações.
4. (11 pontos) O Borzou é dono de todas as estações.
5. (12 pontos) Existe exatamente uma estação de carregamento.
6. (51 pontos) Sem restrições adicionais.

Avaliador de exemplo

O avaliador de exemplo lê o *input* no formato seguinte:

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

O avaliador de exemplo imprime o valor de retorno de `who_wins` no formato seguinte:

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