



Trenino giocattolo

I fratelli gemelli Arezou e Borzou hanno ricevuto per il loro compleanno un magnifico set da ferromodellismo, con cui hanno costruito una rete di n stazioni (numerata da 0 e $n - 1$) ed m binari *unidirezionali*. Ogni binario parte da una stazione ed arriva nella stessa o in un'altra stazione, e c'è sempre **almeno** un binario in uscita da una qualsiasi stazione.

Nella rete c'è un unico trenino che, quando completamente carico, è in grado di percorrere n binari consecutivi (fermandosi prima di entrare nel binario $n + 1$). Alcune delle stazioni sono *stazioni di ricarica*: ogni volta che il trenino arriva in una di queste stazioni si ricarica completamente.

In ogni stazione c'è uno scambio ferroviario che può essere indirizzato **una e una sola volta** su uno qualunque dei binari in uscita così che, quandunque il trenino raggiungerà quella stazione, ne uscirà usando il binario selezionato.

I gemelli vogliono ora fare un gioco con il loro nuovo trenino. Innanzitutto, si sono divisi le stazioni in modo che ogni stazione sia o di Arezou o di Borzou. All'inizio del gioco il trenino viene posizionato nella stazione s con la batteria completamente carica, ed il proprietario di quella stazione ne indirizza lo scambio a suo piacimento su uno qualsiasi dei binari in uscita. A questo punto, il trenino viene acceso ed inizia a muoversi sui binari.

Ogni volta che il trenino entra in una stazione per la prima volta, il suo proprietario ne indirizza lo scambio a suo piacimento; una volta che il binario d'uscita è stato impostato, rimarrà lo stesso per il resto della partita. Quindi, se il trenino rientra in una stazione già visitata, ne uscirà lungo lo stesso binario di prima.

Dato che il numero di stazioni è finito, prima o poi il trenino inizierà necessariamente a seguire un *ciclo* (un ciclo è una sequenza di stazioni *distinte* $c[0], c[1], \dots, c[k - 1]$ tale per cui il trenino esce dalla stazione $c[i]$, $0 \leq i < k - 1$ tramite il binario verso la stazione $c[i + 1]$, ed esce dalla stazione $c[k - 1]$ tramite il binario verso la stazione $c[0]$).

Arezou vince il gioco se il trenino continua a muoversi per sempre, mentre Borzou vince se ad un certo punto il trenino si scarica. In altre parole, se c'è almeno una stazione di ricarica tra le stazioni $c[0], c[1], \dots, c[k - 1]$, il trenino può ricaricarsi e quindi muoversi indefinitamente, e Arezou vince. Se invece questo non avviene, l'energia si esaurirà (possibilmente dopo aver percorso il ciclo diverse volte), e Borzou vince.

Arezou e Borzou giocheranno n partite: nella partita s -esima, con $0 \leq s \leq n - 1$, il trenino inizierà dalla stazione s . Data la descrizione della loro rete ferroviaria, devi capire se per ognuna di queste partite c'è una strategia vincente per Arezou: ovvero, una strategia con la quale vince **indipendentemente** dalle scelte di Borzou.

Dettagli di implementazione

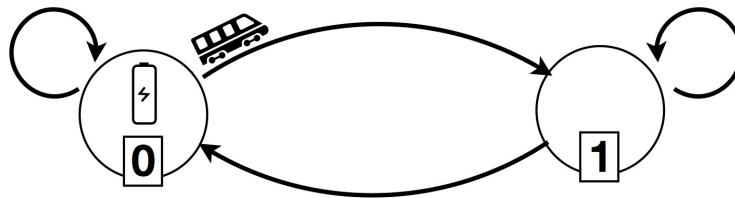
Devi implementare la seguente funzione:

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

- a : un array di lunghezza n per cui $a[i] = 1$ se la stazione i è di Arezou, $a[i] = 0$ se invece è di Borzou (per ogni $0 \leq i < n$).
- r : un array di lunghezza n per cui $r[i] = 1$ se la stazione i è di ricarica, $r[i] = 0$ altrimenti (per ogni $0 \leq i < n$).
- u, v : due array di lunghezza m , tali per cui c'è un binario unidirezionale che parte dalla stazione $u[i]$ ed arriva alla stazione $v[i]$ (per ogni $0 \leq i \leq m - 1$).
- La funzione deve restituire un array w di lunghezza n , tale per cui $w[i] = 1$ se Arezou ha una strategia vincente per la partita in cui il treno parte dalla stazione i , altrimenti $w[i] = 0$ (per ogni $0 \leq i \leq n - 1$).

Esempio

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



- Ci sono 2 stazioni. Borzou possiede la stazione 0, che è di ricarica. Arezou possiede la stazione 1, che non è di ricarica.
- Ci sono 4 binari: $(0, 0)$, $(0, 1)$, $(1, 0)$, e $(1, 1)$, dove (i, j) denota un binario unidirezionale dalla stazione i alla stazione j .
- Consideriamo la partita in cui il trenino inizia dalla stazione 0. Se Borzou indirizza lo scambio della stazione 0 sul binario $(0, 0)$, il trenino percorrerà indefinitamente questo binario (dato che la stazione 0 è di ricarica) facendo vincere Arezou. Se invece Borzou indirizza lo scambio della stazione 0 sul binario $(0, 1)$, Arezou può indirizzare lo scambio della stazione 1 sul binario $(1, 0)$. Anche in questo caso Arezou vince, dato che il trenino circolerà indefinitamente attraverso entrambe le stazioni (di cui una è di ricarica). Complessivamente, Arezou può vincere il gioco indipendentemente dalle scelte di Borzou.
- Con un ragionamento analogo, anche nella partita che inizia dalla stazione 1 Arezou vince indipendentemente dalle scelte di Borzou. Quindi, la funzione deve restituire l'array $[1, 1]$.

Constraints

- $1 \leq n \leq 5000$.
- $n \leq m \leq 20\,000$.
- C'è sempre almeno una stazione di ricarica.
- C'è sempre almeno un binario in uscita da ogni stazione.
- Ci possono essere binari che partono e arrivano nella stessa stazione, per cui $u[i] = v[i]$.
- Ogni binario è distinto, cioè non esistono due indici i e j ($0 \leq i < j \leq m - 1$) per cui $u[i] = u[j]$ e $v[i] = v[j]$.
- $0 \leq u[i], v[i] \leq n - 1$ (per ogni $0 \leq i \leq m - 1$).

Assegnazione del punteggio

1. **(5 punti)** $v[i] = u[i]$ oppure $v[i] = u[i] + 1$ per ogni $0 \leq i \leq m - 1$.
2. **(10 punti)** $n \leq 15$.
3. **(11 punti)** Arezou possiede tutte le stazioni.
4. **(11 punti)** Borzou possiede tutte le stazioni.
5. **(12 punti)** C'è esattamente una stazione di ricarica.
6. **(51 punti)** Nessuna limitazione specifica.

Grader di prova

Il grader di prova legge l'input nel seguente formato::

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

Il grader di prova stampa il valore restituito da `who_wins` nel seguente formato:

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