# IOI '17 P3 - Toy Train

**Time limit:** 1.0s **Memory limit:** 256M

Arezou and her brother Borzou are twins. They have received an amazing toy train set for their birthday, and they used it to build a railway system with $n$ stations and $m$ *one-way* tracks. The stations are numbered from $0$ to $n-1$. Each track starts at one station and ends at the same or a different station. There is at least one track starting at each station.

Some stations are *charging stations*. Whenever the train arrives at a charging station, it gets fully charged. A fully charged train has enough energy to travel along $n$ consecutive tracks. That is, the train runs out of energy just when it enters the $(n+1)^{st}$ track after last being charged.

On each station there is a switch that can be pointed to any of the tracks that start at that station. When a train is at a station, it leaves it using the track that is pointed to by the switch on that station.

The twins are going to play a game with their train. They have already divided all the stations between themselves: each station is either owned by Arezou or by Borzou. There is a single train. At the beginning of the game the train is at station $s$ and it is fully charged. To start the game, the owner of station $s$ points the switch on station $s$ to one of the tracks that start at station $s$. Then, they turn the train on and the train starts traveling along the tracks.

Whenever the train enters a station for the first time, the owner of that station sets the switch on that station. Once a switch is set, it stays in the same position for the rest of the game. Thus, if a train re-enters a station it visited before, it will leave that station along the same track as before.

Since there is a finite number of stations, the train will eventually start going along a *cycle*. A cycle is a sequence of *distinct* stations $c[0], c[1], \ldots, c[k-1]$ such that the train leaves station $c[i]$ (for $0 \le i < k-1$) using a track going to station $c[i+1]$, and it leaves station $c[k-1]$ using a track going to station $c[0]$. Note that a cycle may consist of a single station (i.e., have $k=1$) if the train leaves the station $c[0]$ using a track that goes back to $c[0]$.

Arezou wins the game if the train continues going indefinitely, and Borzou wins if the train runs out of energy. In other words, if there is at least one charging station among $c[0], c[1], \ldots, c[k-1]$, the train can recharge and cycle indefinitely, and Arezou wins. Otherwise, it will run out of energy (possibly after turning around the cycle several times), and Borzou wins.

You are given the description of the railway system. Arezou and Borzou are going to play $n$ games. In the $s^{th}$ game, for $0 \le s \le n-1$, the train will initially be at station $s$. Your task is to find, for each game, whether there is a strategy for Arezou that guarantees she wins, regardless of how Borzou plays.

## Implementation details

You should implement the following procedure:

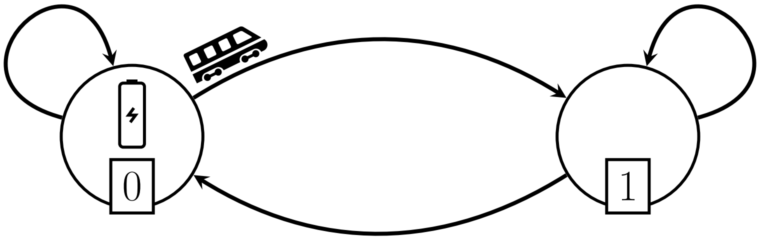
```
std::vector<int> who_wins(std::vector<int> a, std::vector<int> r, std::vector<int> u,
std::vector<int> v)
```

- $a$: array of length $n$. If Arezou owns station $i$, $a[i]=1$. Otherwise, Borzou owns station $i$ and $a[i]=0$.

- $r$: array of length $n$. If the station $i$ is a charging station, $r[i] = 1$. Otherwise, $r[i] = 0$.
- $u$ and $v$: arrays of length $m$. For all $0 \le i \le m - 1$, there is a one-way track starting at station $u[i]$ and ending at station $v[i]$.
- This procedure should return an array $w$ of length $n$. For each $0 \le i \le n - 1$, the value of $w[i]$ should be $1$ if Arezou can win the game that starts at station $i$, regardless of how Borzou plays. Otherwise, the value of $w[i]$ should be $0$.

## Example

```
who_wins({0, 1}, {1, 0}, {0, 0, 1, 1}, {0, 1, 0, 1})
```



- There are $2$ stations. Borzou is the owner of station $0$, which is a charging station. Arezou is the owner of station $1$, which is not a charging station.
- There are $4$ tracks $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$, where $(i, j)$ denotes a one-way track from station $i$ to station $j$.
- Consider the game in which the train is initially placed at station $0$. If Borzou sets the switch on station $0$ towards the track $(0, 0)$, the train will indefinitely cycle through this track (note that station $0$ is a charging station). In this case, Arezou wins. Otherwise, if Borzou sets the switch on station $0$ towards track $(0, 1)$, Arezou can set the switch on station $1$ towards $(1, 0)$. If this happens, the train will indefinitely cycle through both stations. Again Arezou wins, since station $0$ is a charging station and the train will not stop. Hence, Arezou can win the game, regardless of what Borzou does.
- By a similar reasoning, in the game starting at station $1$ Arezou can also win, regardless of how Borzou plays. Thus, the procedure should return $[1, 1]$.

## Constraints

- $1 \le n \le 5000$.
- $n \le m \le 20\,000$.
- There is at least one charging station.
- There is at least one track starting at each station.
- There might be tracks that start and end at the same station (i.e., $u[i] = v[i]$).
- Each track is distinct. In other words, there are no such two indices $i$ and $j$ ($0 \le i < j \le m - 1$) that $u[i] = u[j]$ and $v[i] = v[j]$.
- $0 \le u[i], v[i] \le n - 1$ (for all $0 \le i \le m - 1$).

## Subtasks

1. (5 points) For all $0 \leq i \leq m-1$, either $v[i] = u[i]$ or $v[i] = u[i] + 1$.
2. (10 points) $n \leq 15$.
3. (11 points) Arezou owns all stations.
4. (11 points) Borzou owns all stations.
5. (12 points) There is exactly one charging station.
6. (51 points) No additional constraints.

## Sample grader

The sample grader reads the input in the following format:

- line $1$: $n \; m$
- line $2$: $a[0] \; a[1] \; \ldots \; a[n-1]$
- line $3$: $r[0] \; r[1] \; \ldots \; r[n-1]$
- line $4+i$ (for $0 \leq i \leq m-1$): $u[i] \; v[i]$

The sample grader prints the return value of `who_wins` in the following format:

- line $1$: $w[0] \; w[1] \; \ldots \; w[n-1]$