



Игрушечный поезд

Арезу и Борзу — близнецы. На день рождения им подарили восхитительную игрушечную железную дорогу, из которой они построили железнодорожную сеть, состоящую из n станций и m односторонних путей. Станции пронумерованы от 0 до $n - 1$. Каждый путь выходит из одной станции и приходит либо к той же станции, либо к некоторой другой. С каждой станции выходит хотя бы один путь.

Некоторые станции являются *зарядными*. Когда поезд прибывает на зарядную станцию, он полностью заряжается. Энергии полностью заряженного поезда хватает, чтобы проехать n путей без дополнительной зарядки. Иными словами, поезд полностью разряжается в момент перехода на $(n + 1)$ -й по счёту путь с момента последней зарядки.

На каждой станции имеется переключатель, который может указывать на любой из путей, выходящий от станции. Поезд может покинуть станцию только по пути, на который указывает переключатель.

Близнецы хотят сыграть в игру, используя свой поезд. Они поделили между собой все станции: каждая станция принадлежит либо Арезу, либо Борзу. У них есть один поезд. В начале игры полностью заряженный поезд находится на станции s . Игра начинается с того, что владелец станции s устанавливает переключатель на один из исходящих путей. После этого дети включают поезд и он начинает двигаться вдоль путей.

Когда поезд первый раз оказывается на некоторой станции, владелец станции устанавливает переключатель, расположенный на этой станции. Установленный переключатель остаётся в том же положении до конца игры. Таким образом, если поезд оказывается на станции, где он уже был, то он отправляется с неё по тому же пути, что и раньше.

Так как станций конечное количество, рано или поздно поезд попадёт в *цикл*. Цикл это такая последовательность *различных* станций $c[0], c[1], \dots, c[k - 1]$, что со станции $c[i]$ (для $0 \leq i < k - 1$) поезд выезжает по пути до станции $c[i + 1]$, а со станции $c[k - 1]$ поезд выезжает по пути до станции $c[0]$. Обратите внимание, что цикл может состоять из единственной станции (то есть, $k = 1$), если поезд со станции $c[0]$ выезжает по пути, ведущему обратно на станцию $c[0]$.

Арезу выигрывает игру, если поезд продолжает движение неограниченно долгое время, а Борзу выигрывает, если поезд рано или поздно разрядится. Иными словами, если среди станций $c[0], c[1], \dots, c[k - 1]$ есть хотя бы одна зарядная станция, то поезд сможет постоянно заряжаться и неограниченно долго перемещаться вдоль данного цикла, и Арезу выигрывает. В противном случае поезд разрядится (возможно, после некоторого количества полных перемещений вдоль цикла), и выигрывает Борзу.

Задано описание железнодорожной сети. Арезу и Борзу собираются провести n игр. В s -й игре для $0 \leq s \leq n - 1$ поезд исходно находится на станции s . Для каждой из игр требуется определить, есть ли у Арезу стратегия, гарантирующая ей выигрыш вне зависимости от ходов Борзу.

Детали реализации

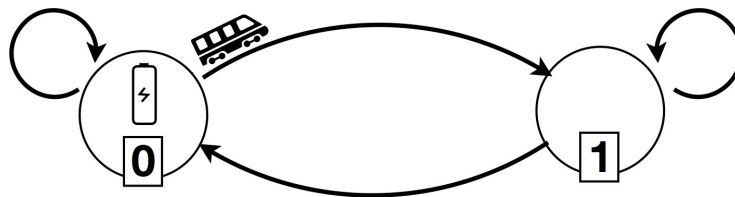
Требуется реализовать следующую функцию (метод):

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

- a : массив длины n . Если станция i принадлежит Арезу, то $a[i] = 1$. В противном случае станция i принадлежит Борзу и $a[i] = 0$.
- r : массив длины n . Если станция i является зарядной, то $r[i] = 1$. В противном случае будет выполнено $r[i] = 0$.
- u и v : массивы длины m , задающие для каждого $0 \leq i \leq m - 1$ односторонний путь, ведущий от станции $u[i]$ к станции $v[i]$.
- Данная функция должна возвращать массив w длины n . Для каждого $0 \leq i \leq n - 1$ значение $w[i]$ должно равняться 1, если Арезу выигрывает в игре, начинающейся со станции i , вне зависимости от ходов Борзу. В противном случае значение $w[i]$ должно быть установлено в 0.

Пример

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



- В железнодорожной сети есть две станции. Борзу принадлежит станция 0, которая является зарядной. Арезу принадлежит станция 1, которая не является зарядной.
- В сети есть 4 пути $(0, 0)$, $(0, 1)$, $(1, 0)$, и $(1, 1)$, где (i, j) обозначает односторонний путь, идущий от станции i к станции j .
- Рассмотрим игру, в которой поезд исходно расположен на станции 0. Если Борзу установит переключатель станции 0 на путь $(0, 0)$, поезд будет неограниченно перемещаться по циклу, состоящему из этого пути (обратите внимание, что станция 0 является зарядной). В этом случае выиграет Арезу. В противном случае, если Борзу установит переключатель станции 0 на путь $(0, 1)$, Арезу может установить переключатель станции 1 на путь $(1, 0)$. Поезд будет неограниченно перемещаться между двумя станциями. Арезу выиграет, так как станция 0 является зарядной, и поезд

никогда не остановится. Таким образом, Арезу может выиграть игру вне зависимости от ходов Борзу.

- Аналогично можно убедиться, что в игре, начинающейся на станции 1, Арезу также выигрывает, вне зависимости от ходов Борзу. Таким образом, функция должна вернуть $[1, 1]$.

Ограничения

- $1 \leq n \leq 5000$.
- $n \leq m \leq 20\,000$.
- Имеется хотя бы одна зарядная станция.
- Из каждой станции выходит хотя бы один путь.
- Некоторые пути могут начинаться и заканчиваться в одной и той же станции (то есть, $u[i] = v[i]$).
- Все пути различаются. Иными словами, не существуют двух таких индексов i и j ($0 \leq i < j \leq m - 1$), что $u[i] = u[j]$ и $v[i] = v[j]$.
- $0 \leq u[i], v[i] \leq n - 1$ (для всех $0 \leq i \leq m - 1$).

Система оценивания

1. (5 баллов) Для всех $0 \leq i \leq m - 1$ выполнено либо $v[i] = u[i]$, либо $v[i] = u[i] + 1$.
2. (10 баллов) $n \leq 15$.
3. (11 баллов) Все станции принадлежат Арезу.
4. (11 баллов) Все станции принадлежат Борзу.
5. (12 баллов) Ровно одна станция является зарядной.
6. (51 балл) Дополнительные ограничения отсутствуют.

Пример проверяющего модуля

Пример проверяющего модуля читает входные данные в следующем формате:

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

Пример проверяющего модуля выводит массив, возвращаемый из функции `who_wins`, в следующем формате:

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