



Thousands Islands

Тысячи Островов это группа красивых островов в Яванском море. Она состоит из N островов, с номерами от 0 до $N - 1$.

Есть M каноэ, с номерами от 0 до $M - 1$, которые можно использовать для перемещения с острова на остров. Для каждого i , такого что $0 \leq i \leq M - 1$, каноэ i может быть пришвартовано к острову $U[i]$ или к острову $V[i]$, и может быть использовано для перемещения между островами $U[i]$ и $V[i]$. А именно, когда каноэ пришвартовано к острову $U[i]$, его можно использовать для перемещения с острова $U[i]$ на остров $V[i]$, после чего каноэ становится пришвартованным к острову $V[i]$. Аналогично, когда каноэ пришвартовано к острову $V[i]$, его можно использовать для перемещения с острова $V[i]$ на остров $U[i]$, после чего каноэ становится пришвартованным к острову $U[i]$. Изначально каноэ пришвартовано к острову $U[i]$. Несколько каноэ могут использоваться для перемещения между одной и той же парой островов. На одном острове допускается швартовать несколько каноэ.

Из соображений безопасности, каноэ нуждается в профилактическом осмотре после каждого перемещения, поэтому одно и то же каноэ запрещается использовать два раза подряд. Таким образом, после использования каноэ i , необходимо использовать другое каноэ, прежде чем каноэ i станет снова доступным.

Бу Денгклек планирует проложить маршрут по некоторым из островов. Её маршрут **допустим** если и только если удовлетворены следующие условия.

- Она начинает и заканчивает маршрут на острове 0.
- Она посещает как минимум один остров, отличный от острова 0.
- После окончания маршрута каждое каноэ пришвартовано к тому же острову, что и до начала маршрута. То есть каноэ i , для каждого i , такого что $0 \leq i \leq M - 1$, должно быть пришвартовано к острову $U[i]$.

Помогите Бу Денгклек найти любой допустимый маршрут, включающий не более 2 000 000 перемещений, или определите, что допустимого маршрута не существует. Можно доказать, что для ограничений, указанных в задаче (смотри секцию Constraints), если допустимый маршрут существует, то также существует допустимый маршрут, который включает не больше 2 000 000 перемещений.

Implementation Details

Вы должны реализовать следующую функцию:

```
union(bool, int[]) find_journey(int N, int M, int[] U, int[] V)
```

- N : количество островов.
- M : количество каноэ.
- U, V : массивы длины M , описывающие каноэ.
- Эта функция должна вернуть или логическое значение, или массив целых чисел.
 - Если допустимого маршрута не существует, функция должна вернуть `false`.
 - Если допустимый маршрут существует, у вас есть два варианта:
 - Чтобы получить полный балл, функция должна вернуть массив из не более чем 2 000 000 целых чисел, задающий допустимый маршрут. Элементами этого массива должны быть номера каноэ, используемые в маршруте (в том порядке, в котором их использовали).
 - Чтобы получить частичный балл, функция должна вернуть `true`, массив из более чем 2 000 000 чисел или массив, который не задает допустимый маршрут (см. раздел Subtasks)
- Эта функция вызывается ровно один раз.

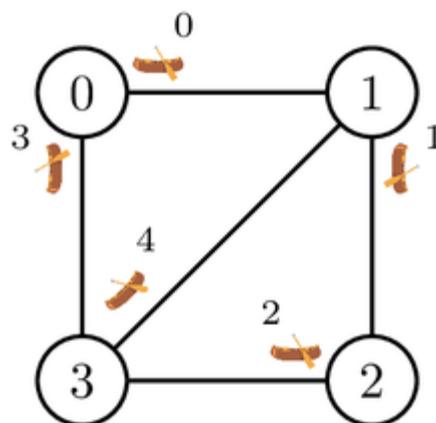
Examples

Example 1

Рассмотрим следующий вызов:

```
find_journey(4, 5, [0, 1, 2, 0, 3], [1, 2, 3, 3, 1])
```

Острова и каноэ показаны на изображении ниже.



Один из возможных допустимых маршрутов следующий. Бу Денгклек сначала перемещается на каноэ 0, 1, 2, и 4 в этом порядке. В результате она на острове 1. После этого, Бу Денгклек может использовать каноэ 0 снова, так как оно в текущий момент пришвартовано к острову 1, и последнее использованное каноэ не 0. После перемещения на каноэ 0, Бу Денгклек теперь на острове 0. Однако, каноэ 1, 2 и 4 пришвартованы не к тем островам, как было перед началом маршрута. Бу Денгклек продолжает маршрут, перемещаясь на каноэ 3, 2, 1, 4, и 3 ещё раз. Бу Денгклек вернулась на остров 0 и все каноэ пришвартованы к тем же островам, что и до начала прохождения маршрута.

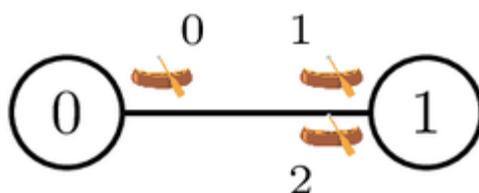
Таким образом, $[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]$ является допустимым массивом, который может вернуть функция.

Example 2

Рассмотрим следующий вызов:

```
find_journey(2, 3, [0, 1, 1], [1, 0, 0])
```

Острова и каноэ показаны на изображении ниже.



Бу Денгклек может начать маршрут только с каноэ 0, после чего она может использовать каноэ 1 или 2. Заметим, что она не может использовать каноэ 0 два раза подряд. В обоих случаях, Бу Денгклек снова на острове 0. Однако, каноэ пришвартованы не к тем островам, как было перед началом маршрута, и Бу Денгклек не может продолжить маршрут, так как единственное каноэ, пришвартованное на острове 0, это то, которое она только что использовала. Так как нет допустимого маршрута, функция должна вернуть `false`.

Constraints

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 200\,000$
- $0 \leq U[i] \leq N - 1$ и $0 \leq V[i] \leq N - 1$ (для каждого i , такого что $0 \leq i \leq M - 1$)
- $U[i] \neq V[i]$ (для каждого i , такого что $0 \leq i \leq M - 1$)

Subtasks

1. (5 баллов) $N = 2$
2. (5 баллов) $N \leq 400$. Для каждой пары различных островов x и y ($0 \leq x < y \leq N - 1$), существует ровно два каноэ, которые могут быть использованы для перемещения между ними. Одно из них пришвартовано к острову x , а другое к острову y .
3. (21 балл) $N \leq 1000$, M чётно, и для каждого **чётного** i , такого что $0 \leq i \leq M - 1$, каноэ i и $i + 1$ оба могут использоваться для перемещения между островами $U[i]$ и $V[i]$. Каноэ i изначально пришвартовано к острову $U[i]$, а каноэ $i + 1$ изначально пришвартовано к острову $V[i]$. Формально, $U[i] = V[i + 1]$ и $V[i] = U[i + 1]$.
4. (24 балла) $N \leq 1000$, M чётно, и для каждого **чётного** i , такого что $0 \leq i \leq M - 1$, каноэ i и $i + 1$ оба могут использоваться для перемещения между островами $U[i]$ и $V[i]$. Оба каноэ изначально пришвартованы к острову $U[i]$. Формально, $U[i] = U[i + 1]$ и $V[i] = V[i + 1]$.
5. (45 баллов) Нет дополнительных ограничений.

Для каждого набора тестовых данных, в котором существует допустимый маршрут, ваше решение:

- получает полный балл, если возвращает допустимый маршрут,
- получает 35% баллов если возвращает true, массив более чем из 2 000 000 целых чисел, или массив из целых чисел, не описывающий допустимый маршрут,
- получает 0 баллов во всех остальных случаях.

Для каждого набора тестовых данных, в котором не существует допустимый маршрут, ваше решение:

- получает полный балл, если возвращает false,
- получает 0 баллов во всех остальных случаях.

Обратите внимание, что итоговая оценка каждой подзадачи равна минимуму баллов для всех тестовых наборов в подзадаче.

Sample Grader

Доступный вам грайдер считывает данные в следующем формате:

- строка 1: $N M$
- строка $2 + i$ ($0 \leq i \leq M - 1$): $U[i] V[i]$

Доступный вам грайдер выводит ваши ответы в следующем формате:

- Если `find_journey` возвращает bool:
 - строка 1: 0
 - строка 2: 0 если `find_journey` возвращает false, или 1 в противном случае.

- Если `find_journey` возвращает `int[]`, обозначим элементы этого массива $c[0], c[1], \dots, c[k-1]$ соответственно. Доступный вам грейдер выводит:
 - строка 1: 1
 - строка 2: k
 - строка 3: $c[0] \ c[1] \ \dots \ c[k-1]$