



Хилядите острови

Хилядите острови е група от красиви острови, разположени в Яванско море. Тя се състои се от N острова, номерирани от 0 до $N - 1$.

Има M канута, номерирани от 0 до $M - 1$, които могат да се използват за плаване между островите. За всяко i , такава че $0 \leq i \leq M - 1$, кануто i може да акостира на остров $U[i]$ или на остров $V[i]$ и може да се използва за плаване между островите $U[i]$ и $V[i]$. По-конкретно, когато кануто е акостира на остров $U[i]$, то може да се използва за плаване от остров $U[i]$ до остров $V[i]$, след което кануто акостира на остров $V[i]$. По същия начин, когато кануто е акостира на остров $V[i]$, то може да се използва за плаване от остров $V[i]$ до остров $U[i]$, след което кануто акостира на остров $U[i]$. Първоначално кануто е акостира на остров $U[i]$. Възможно е няколко канута да могат да се използват за плаване между една и съща двойка острови. Също така е възможно няколко канута да са акостирали на един и същ остров.

От съображения за безопасност едно кану трябва да премине поддръжка след всяко плаване с него, което забранява да се плава два пъти подред с едно и също кану. Тоест, след използване на някое кану i , трябва да се използва друго кану, преди кану i да може да се използва отново.

Бу Денглек иска да планира пътуване през някои от островите. Пътуването ѝ е **валидно**, тогава и само тогава, когато са изпълнени следните условия.

- Тя започва и завършва своето пътуване на остров 0.
- Тя посещава поне един остров, различен от остров 0.
- След като пътуването приключи, всяко кану е акостира на същия остров, както е било преди пътуването. Т.е. кану i , за всяко i , такава че $0 \leq i \leq M - 1$, трябва да е акостира на остров $U[i]$.

Помогнете на Бу Денглек да намери някакво валидно пътуване, включващо най-много 2 000 000 плавания, или установете, че не съществува пътуване, отговарящо на условията за валидност. Може да се докаже, че при ограниченията, посочени в тази задача (вижте раздела Ограничения), ако съществува валидно пътуване, то съществува и валидно пътуване, което не включва повече от 2 000 000 плавания.

Детайли по имплементацията

Трябва да имплементирате следната процедура:

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

- N : броят на островите.
- M : броят на канутата.
- U, V : масиви с дължина M , описващи канутата.
- Тази процедура трябва да върне или булева стойност, или масив от цели числа.
 - Ако не съществува валидно пътуване, процедурата трябва да върне `false`.
 - Ако съществува валидно пътуване, имате две възможности:
 - За да получите пълния брой точки, процедурата трябва да върне масив от най-много 2 000 000 цели числа, представляващи валидно пътуване. Точно, елементите на този масив трябва да бъдат номерата на канутата, които се използват в пътуването (в реда, в който се използват).
 - За да получите частичен резултат, процедурата трябва да върне `true`, масив от повече от 2 000 000 цели числа или масив от цели числа, които не описват валидно пътуване. (За повече подробности вижте раздела Подзадачи.)
- Тази процедура се извиква точно веднъж.

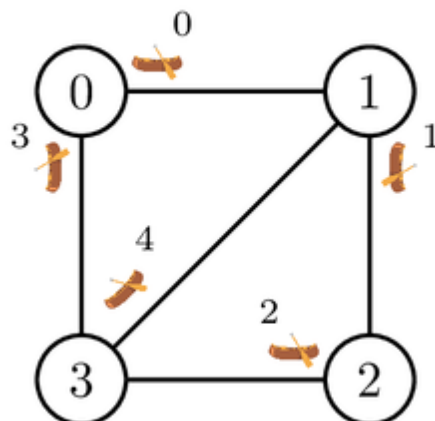
Примери

Пример 1

Разгледайте следното извикване:

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

Островите и канутата са показани на фигурата по-долу.



Едно възможно валидно пътуване е следното. Бу Денгклек първо плава с канута 0, 1, 2 и 4 в този ред. В резултат на това тя е на остров 1. След това Бу Денгклек може отново да плава с

кану 0, тъй като в момента то е акостирали на остров 1 и последното кану, което е използвала, не е кану 0. След като отново плава с кану 0, Бу Денгклек вече е на остров 0. Въпреки това канутата 1, 2 и 4 не са акостирали на същите острови, както преди пътуването. След това Бу Денгклек продължава пътуването си с канута 3, 2, 1, 4 и отново 3. Бу Денгклек се връща на остров 0 и всички канута са акостирали на същите острови, както преди пътуването.

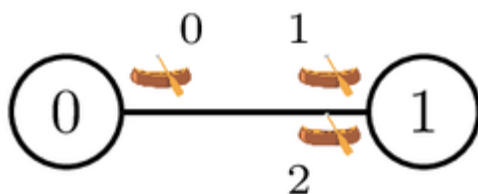
Следователно, върнатата стойност $[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]$ представлява валидно пътуване.

Пример 2

Разгледайте следното извикване:

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

Островите и канутата са показани на фигурата по-долу.



Бу Денгклек може да започне само с плаване с кану 0, след което може да плава с кану 1 или 2. Имайте предвид, че тя не може да плава с кану 0 два пъти подред. И в двата случая Бу Денгклек се връща на остров 0. Въпреки това, канутата не са акостирали на същите острови, както преди пътуването, и Бу Денгклек не може да плава с кану след това, тъй като единственото кану, акостирали на остров 0, е това, което току-що е използвано. Тъй като няма валидно пътуване, процедурата трябва да върне `false`.

Ограничения

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 200\,000$
- $0 \leq U[i] \leq N - 1$ и $0 \leq V[i] \leq N - 1$ (за всяко i , такова че $0 \leq i \leq M - 1$)
- $U[i] \neq V[i]$ (за всяко i , такова че $0 \leq i \leq M - 1$)

Подзадачи

1. (5 точки) $N = 2$

2. (5 точки) $N \leq 400$. За всяка двойка различни острови x и y ($0 \leq x < y \leq N - 1$) има точно две канута, които могат да се използват за плаване между тях. Едното от тях първоначално е акостирано на остров x , а другото - на остров y .
3. (21 точки) $N \leq 1000$, M е четно, и за всяко **четно** i , такава че $0 \leq i \leq M - 1$, канутата i и $i + 1$ могат да бъдат използвани за плаване между острови $U[i]$ и $V[i]$. Кану i е първоначално акостирано на остров $U[i]$, а кану $i + 1$ е първоначално акостирано на остров $V[i]$. Формално, $U[i] = V[i + 1]$ и $V[i] = U[i + 1]$.
4. (24 точки) $N \leq 1000$, M е четно, и за всяко **четно** i , такава че $0 \leq i \leq M - 1$, канутата i и $i + 1$ могат да бъдат използвани за плаване между острови $U[i]$ и $V[i]$. И двете канута първоначално са акостирали на остров $U[i]$. Формално, $U[i] = U[i + 1]$ and $V[i] = V[i + 1]$.
5. (45 точки) Без допълнителни ограничения.

За всеки тестов случай, в който съществува валидно пътуване, вашето решение:

- получава пълните точки, ако върне валидно пътуване,
- получава 35% от точките, ако върне true, масив от повече от 2 000 000 цели числа или масив, които не описва валидно пътуване,
- получава 0 точки в противен случай.

За всеки тестов случай, в който не съществува валидно пътуване, вашето решение:

- получава пълните точки, ако върне false,
- получава 0 точки в противен случай.

Обърнете внимание, че крайният резултат за всяка подзадача е минимумът от броя точки за тестовите случаи в подзадачата.

Примерен грейдър

Примерният грейдър прочита входа в следния формат:

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

Примерният грейдър отпечатва вашите отговори в следния формат:

- Ако `find_journey` върне стойност от тип `bool`:
 - ред 1: 0
 - ред 2: 0, ако `find_journey` върне `false`, или 1 в противен случай.
- Ако `find_journey` върне масив от тип `int []`, означаваме елементите на този масив с $c[0], c[1], \dots, c[k - 1]$. Примерният грейдър отпечатва:
 - ред 1: 1
 - ред 2: k
 - ред 3: $c[0] c[1] \dots c[k - 1]$