Thousands Islands

Vạn Đảo là một nhóm những hòn đảo tuyệt đẹp nằm ở biển Java. Có tổng cộng N hòn đảo, được đánh số từ 0 đến $N-1$.

Có M ca nô, được đánh số từ 0 đến $M-1$, được sử dụng để di chuyển giữa các hòn đảo. Với mỗi i mà $0 \leq i \leq M-1$, ca nô i có thể neo đậu ở đảo $U[i]$ hoặc $V[i]$, và được sử dụng để di chuyển giữa đảo $U[i]$ và $V[i]$. Cụ thể, khi ca nô đang neo đậu ở đảo $U[i]$, nó có thể được sử dụng để di chuyển từ đảo $U[i]$ đến đảo $V[i]$, sau đó ca nô sẽ được neo đậu tại đảo $V[i]$. Tương tự, khi ca nô đang được neo đậu ở đảo $V[i]$, nó có thể được sử dụng để di chuyển từ đảo $V[i]$ đến đảo $U[i]$, sau đó ca nô sẽ được neo đậu tại đảo $U[i]$. Ban đầu, ca nô được neo đậu tại đảo $U[i]$. Có thể có nhiều ca nô được sử dụng để di chuyển giữa một cặp hai hòn đảo. Cũng có thể có nhiều ca nô cùng được neo đậu tại một hòn đảo.

Vì lý do an toàn, mỗi chiếc ca nô cần phải được bảo dưỡng sau mỗi lần nó di chuyển, điều này làm cho một chiếc ca nô không được phép di chuyển hai lần liên tiếp. Nói cách khác, sau khi sử dụng ca nô i, một chiếc ca nô khác phải được sử dụng trước khi ca nô i có thể được sử dụng tiếp.

Bu Dengklek muốn lên kế hoạch cho một hành trình quanh một số hòn đảo. Một hành trình là **hợp lệ** khi và chỉ khi thoả mãn những điều kiện sau.

- Cô ấy bắt đầu và kết thúc hành trình ở hòn đảo 0.
- Cô phải ghé thăm ít nhất một hòn đảo khác hòn đảo 0.
- Sau khi hành trình kết thúc, mỗi chiếc ca nô phải được neo đậu tại hòn đảo mà nó đã được neo đậu trước hành trình. Có nghĩa là, ca nô i, với mỗi i mà $0 \leq i \leq M-1$, phải được neo đậu tại đảo $U[i]$.

Hãy giúp Bu Dengklek tìm ra một hành trình hợp lệ gồm nhiều nhất $2\,000\,000$ lần di chuyển, hoặc xác định rằng không có một hành trình hợp lệ nào tồn tại. Có thể chứng minh được rằng với những ràng buộc trong bài toán này (xem phần Ràng buộc), nếu có một hành trình hợp lệ bất kỳ tồn tại, thì cũng có một hành trình hợp lệ mà không bao gồm quá $2\,000\,000$ lần di chuyển.

Chi tiết cài đặt

Bạn cần phải cài đặt hàm sau:

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

- N: số lượng hòn đảo.
- M: số lượng ca nô.
- U, V: các mảng gồm M phần tử mô tả các ca nô.
- Hàm cần trả về một biến boolean hoặc một mảng các số nguyên.
    - Nếu không có hành trình hợp lệ, hàm cần trả về giá trị `false`.
    - Nếu tồn tại một hành trình hợp lệ, có hai lựa chọn:
        - Để nhận được điểm tuyệt đối, hàm cần trả về một mảng nhiều nhất $2\,000\,000$ số nguyên thể hiện hành trình hợp lệ đó. Cụ thể hơn, các phần tử của mảng này phải là các chỉ số của các ca nô được sử dụng trong hành trình (theo thứ tự mà chúng được sử dụng).
        - Để nhận được một phần điểm, hàm cần trả về `true`, một mảng nhiều hơn $2\,000\,000$ số nguyên, hay một mảng số nguyên không thể hiện một hành trình hợp lệ. (Xem chi tiết ở phần Subtask).
- Hàm này được gọi đúng một lần.

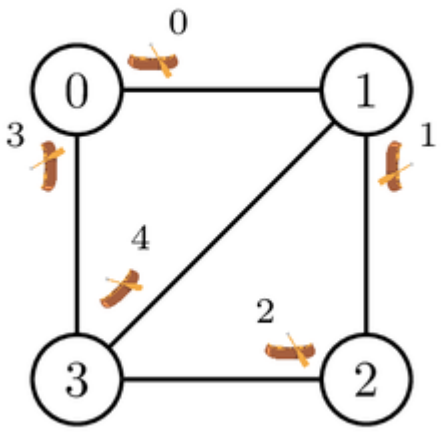
Ví dụ

Ví dụ 1

Xét lời gọi hàm:

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

Các đảo và các ca nô được mô tả ở bức ảnh sau đây.



Một hành trình hợp lệ có thể là. Đầu tiên, Bu Dengklek lần lượt sử dụng các ca nô 0, 1, 2 và 4. Lúc này, cô ấy ở đảo 1. Sau đó, Bu Dengklek có thể sử dụng ca nô 0 một lần nữa vì lúc này nó đang được neo đậu ở đảo 1 và ca nô trước đó mà cô ấy sử dụng không phải là ca nô 0. Sau khi di chuyển bằng ca nô 0 một lần nữa, Bu Dengklek ở đảo 0. Tuy nhiên, ca nô 1, 2 và 4 đang không được neo đậu ở hòn đảo ban đầu trước hành trình. Bu Dengklek sau đó tiếp tục hành trình bằng cách sử

dụng ca nô 3, 2, 1, 4 và 3 một lần nữa. Bu Dengklek bây giờ đã trở lại đảo 0 và tất cả các ca nô đã được neo đậu ở hòn đảo ban đầu trước khi hành trình bắt đầu.

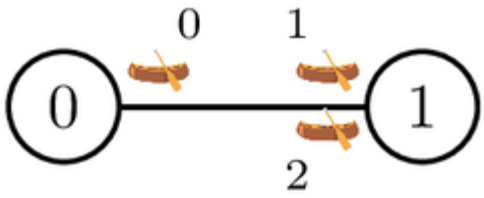
Vì vậy, kết quả trả về $[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]$ thể hiện một hành trình hợp lệ.

Ví dụ 2

Xét lời gọi hàm sau:

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

Các đảo và các ca nô được mô tả ở bức ảnh sau đây.



Bu Dengklek chỉ có thể bắt đầu bằng việc di chuyển với ca nô 0, sau đó cô ấy có thể di chuyển với ca nô 1 hoặc 2. Lưu ý rằng cô ấy không thể di chuyển với ca nô 0 hai lần liên tiếp. Trong cả hai trường hợp, Bu Dengklek đều quay trở lại đảo 0. Tuy nhiên, các ca nô đang không được neo đậu tại hòn đảo ban đầu trước hành trình, và Bu Dengklek không thể di chuyển bằng ca nô nào tiếp theo vì ca nô duy nhất đang ở đảo 0 là chiếc ca nô cô ấy vừa mới sử dụng. Vì không có một hành trình hợp lệ, hàm cần trả về giá trị `false`.

Ràng buộc

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 200\,000$
- $0 \leq U[i] \leq N - 1$ và $0 \leq V[i] \leq N - 1$ (với mỗi i mà $0 \leq i \leq M - 1$)
- $U[i] \neq V[i]$ (với mỗi i mà $0 \leq i \leq M - 1$)

Subtask

1. (5 điểm) $N = 2$
2. (5 điểm) $N \leq 400$. Với mỗi cặp đảo x và y khác nhau ($0 \leq x < y \leq N - 1$), có đúng hai ca nô được sử dụng để di chuyển giữa chúng. Một ca nô được neo đậu ở đảo x, ca nô còn lại được neo đậu ở đảo y.

3. (21 điểm) $N \leq 1000$, M chẵn, với mỗi i **chẵn** mà $0 \leq i \leq M-1$, ca nô i và $i+1$ đều có thể được sử dụng để di chuyển giữa đảo $U[i]$ và đảo $V[i]$. Ban đầu, ca nô i được neo đậu tại đảo $U[i]$ còn ca nô $i+1$ được neo đậu tại đảo $V[i]$. Cụ thể, $U[i] = V[i+1]$ và $V[i] = U[i+1]$.
4. (24 điểm) $N \leq 1000$, M chẵn, với mỗi i **chẵn** mà $0 \leq i \leq M-1$, ca nô i và $i+1$ đều có thể được sử dụng để di chuyển giữa đảo $U[i]$ và đảo $V[i]$. Ban đầu, cả hai ca nô được neo đậu tại đảo $U[i]$. Cụ thể, $U[i] = U[i+1]$ và $V[i] = V[i+1]$.
5. (45 điểm) Không có ràng buộc nào thêm.

Với mỗi trường hợp thử nghiệm mà có tồn tại một hành trình hợp lệ, lời giải của bạn:

- nhận được điểm tối đa nếu bạn trả về một hành trình hợp lệ,
- nhận được 35% số điểm nếu bạn trả về `true`, một mảng có nhiều hơn $2\,000\,000$ số nguyên, hoặc một mảng không mô tả một hành trình hợp lệ,
- nhận được 0 điểm trong các trường hợp khác.

Với mỗi trường hợp thử nghiệm mà không tồn tại một hành trình hợp lệ, lời giải của bạn:

- nhận được điểm tối đa nếu bạn trả về `false`,
- nhận được 0 điểm trong các trường hợp khác.

Lưu ý rằng điểm cho mỗi subtask là số điểm nhỏ nhất của các trường hợp thử nghiệm trong subtask đó.

Trình chấm mẫu

Trình chấm mẫu đọc dữ liệu theo định dạng sau:

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

Trình chấm mẫu in kết quả của bạn theo định dạng sau:

- Nếu hàm `find_journey` trả về một biến kiểu `bool`:
  - dòng 1: 0
  - dòng 2: 0 nếu hàm `find_journey` trả về `false`, hay 1 nếu ngược lại.
- Nếu hàm `find_journey` trả về một mảng `int[]`, gọi các phần tử của mảng này là $c[0], c[1], \ldots c[k-1]$. Trình chấm mẫu sẽ in ra:
  - dòng 1: 1
  - dòng 2: k
  - dòng 3: $c[0]$ $c[1]$ $\ldots$ $c[k-1]$