



## De Duizendeilanden

De Duizendeilanden is een groep prachtige eilanden in de Javazee. Het bestaat uit  $N$  eilanden, genummerd van 0 tot en met  $N - 1$ .

Er zijn  $M$  kano's, genummerd van 0 tot en met  $M - 1$  die gebruikt kunnen worden om tussen eilanden te varen. Kano  $i$ , met  $0 \leq i \leq M - 1$ , kan aangelegd worden bij eilanden  $U[i]$  en  $V[i]$  en kan gebruikt worden om tussen eilanden  $U[i]$  en  $V[i]$  te varen. Om preciezer te zijn, wanneer de kano aangelegd is bij eiland  $U[i]$  kan die gebruikt worden om te varen van eiland  $U[i]$  naar eiland  $V[i]$  waarna die aangelegd wordt bij eiland  $V[i]$ . Op dezelfde manier, als de kano is aangelegd bij eiland  $V[i]$  kan die gebruikt worden om te varen van eiland  $V[i]$  naar eiland  $U[i]$  waarna die aangelegd wordt bij eiland  $U[i]$ . Aan het begin is de kano aangelegd bij eiland  $U[i]$ . Het is mogelijk dat meerdere kano's gebruikt kunnen worden om tussen dezelfde twee eilanden te varen. Het is ook mogelijk dat meerdere kano's aangelegd zijn bij hetzelfde eiland.

Om veiligheidsredenen moet een kano worden onderhouden elke keer nadat er mee wordt gevaren. Het is daarom verboden om dezelfde kano twee keer achter elkaar te gebruiken. Dat betekent dat na het gebruiken van kano  $i$  er eerst een andere kano gebruikt moet worden voordat kano  $i$  weer gebruikt kan worden.

Bu Dengklek wil een reis plannen tussen sommige van deze eilanden. Haar reis is **geldig** dan en slechts dan als aan de volgende voorwaarden wordt voldaan:

- Haar reis start en eindigt op eiland 0.
- Ze bezoekt minstens één ander eiland dan eiland 0.
- Nadat haar reis is afgelopen is elke kano aangelegd bij hetzelfde eiland waar die lag voor de reis begon. In andere woorden, kano  $i$  (met  $0 \leq i \leq M - 1$ ) moet aangelegd worden bij eiland  $U[i]$  aan het eind van de reis.

Help Bu Dengklek met het vinden van een geldige reis waarbij hooguit 2 000 000 keer wordt gevaren, of bepaal dat zo'n reis niet bestaat. Het kan worden bewezen dat onder de randvoorwaarden van deze opgave (zie Randvoorwaarden), dat als er een geldige reis bestaat, er ook een geldige reis is die niet meer dan 2 000 000 keer vaart.

## Implementatiedetails

Je moet de volgende functie implementeren:

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

- $N$ : het aantal eilanden.
- $M$ : het aantal kano's.
- $U, V$ : arrays van lengte  $M$  die de kano's beschrijven
- Deze functie moet een boolean of een array van integers teruggeven.
  - Als er geen geldige reis bestaat moet de functie `false` teruggeven.
  - Als een geldige reis bestaat heb je twee opties:
    - Om een volledige score te krijgen moet de functie een array van hooguit  $2 \cdot 10^6$  integers teruggeven die een geldige reis beschrijft. Om precies te zijn moeten de elementen van deze array de nummers van de kano's zijn in de volgorde dat ze worden gebruikt in de reis.
    - Voor een deelscore moet de functie `true` teruggeven of een array van meer dan  $2 \cdot 10^6$  integers of een array die geen geldige reis beschrijft. Zie Deelopgaven voor meer details.
- Deze functie wordt precies één keer aangeroepen.

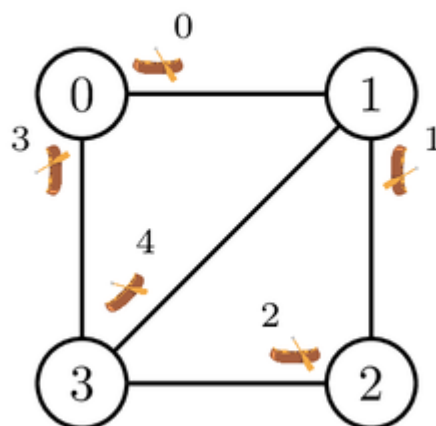
## Voorbeelden

### Voorbeeld 1

Beschouw de volgende aanroep:

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

De eilanden en de kano's worden in het volgende plaatje afgebeeld.



Een mogelijke reis is als volgt. Bu Dengklek vaart eerst in kano's 0, 1, 2 en 4 in die volgorde. Daardoor is ze op eiland 1. Daarna kan Bu Dengklek weer kano 0 gebruiken omdat die dan is aangelegd bij eiland 1 en de laatste kano die ze heeft gebruikt niet kano 0 is. Na het gebruiken van

kano 0 is Bu Dengklek weer bij eiland 0. Echter zijn kano's 1, 2 en 4 nog niet aangelegd bij het eiland waar ze begonnen voor de reis. Bu Dengklek vervolgt haar reis door kano's 3, 2, 1, 4 en 3 weer te gebruiken. Bu Dengklek is weer terug bij eiland 0 en alle kano's zijn weer aangelegd bij dezelfde eilanden als voor haar reis.

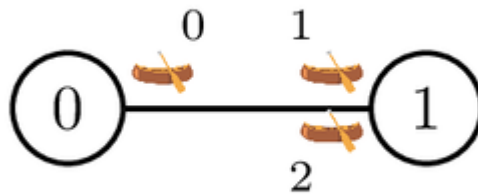
De teruggegeven waarde  $[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]$  is dus een geldige reis.

## Voorbeeld 2

Beschouw de volgende aanroep:

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

De eilanden en kano's zijn afgebeeld in de afbeelding hieronder.



Bu Dengklek kan alleen beginnen met kano 0, waarna ze kan varen met kano 1 of 2. Merk op dat ze niet twee keer achter elkaar met kano 0 kan varen. In beide gevallen is Bu Dengklek weer terug bij eiland 0. Echter zijn de kano's nog niet aangelegd bij de eilanden waar ze begonnen. Bovendien kan Bu Dengklek geen kano meer varen omdat de enige kano die is aangelegd bij eiland 0 de kano is die ze net had gebruikt. Aangezien er geen geldige reis is moet de functie `false` teruggeven.

## Randvoorwaarden

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 200\,000$
- $0 \leq U[i] \leq N - 1$  en  $0 \leq V[i] \leq N - 1$  (voor elke  $i$  met  $0 \leq i \leq M - 1$ )
- $U[i] \neq V[i]$  (voor elke  $i$  met  $0 \leq i \leq M - 1$ )

## Deelopgaven

1. (5 punten)  $N = 2$
2. (5 punten)  $N \leq 400$ . Voor elk paar verschillende eilanden  $x$  en  $y$  ( $0 \leq x < y \leq N - 1$ ) zijn er precies twee kano's die gebruikt kunnen worden om tussen de eilanden te varen. Een van de kano's is aangelegd bij eiland  $x$ , de ander bij eiland  $y$ .

3. (21 punten)  $N \leq 1000$ ,  $M$  is even, en voor elke **even**  $i$  met  $0 \leq i \leq M - 1$ , kano's  $i$  en  $i + 1$  kunnen niet beide gebruikt worden om te varen tussen eilanden  $U[i]$  en  $V[i]$ . Kano  $i$  is in het begin aangelegd bij eiland  $U[i]$  en kano  $i + 1$  is aangelegd bij eiland  $V[i]$ . Formeel:  $U[i] = V[i + 1]$  en  $V[i] = U[i + 1]$ .
4. (24 punten)  $N \leq 1000$ ,  $M$  is even, en voor elke **even**  $i$  met  $0 \leq i \leq M - 1$ , kano's  $i$  en  $i + 1$  kunnen beide gebruikt worden om te varen tussen eilanden  $U[i]$  en  $V[i]$ . Beide kano's zijn in het begin aangelegd bij eiland  $U[i]$ . Formeel:  $U[i] = U[i + 1]$  en  $V[i] = V[i + 1]$ .
5. (45 punten) Geen aanvullende randvoorwaarden

Voor elk testgeval waar een geldige reis bestaat krijgt jouw oplossing:

- alle punten als het een geldige reis teruggeeft.
- 35% van de punten als het een van de volgende waarden teruggeeft: true, een array van meer dan 2 000 000 integers of een array die geen geldige reis beschrijft.
- 0 punten in de andere gevallen.

Voor elk testgeval waarbij een geldige reis niet bestaat krijgt je oplossing:

- volledige punten als het false teruggeeft,
- 0 punten in de andere gevallen.

Merk op dat de eindscore voor elke deelopgave het minimum is van het aantal punten voor de testgevallen van de deelopgave.

## Voorbeeldgrader

De voorbeeldgrader leest de invoer in het volgende formaat:

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

De voorbeeldgrader print je antwoorden in het volgende formaat:

- Als `find_journey` een bool teruggeeft:
  - regel 1: 0
  - regel 2: 0 als `find_journey` false teruggeeft, anders 1.
- Als `find_journey` een `int[]` teruggeeft waarvan we de elementen  $c[0], c[1], \dots, c[k - 1]$  noemen:
  - regel 1: 1
  - regel 2:  $k$
  - regel 3:  $c[0] c[1] \dots c[k - 1]$