



## Thousands Islands

იავას ზღვაში არსებულ კუნძულთა ჯგუფში ათასობით უმშვენიერესი კუნძულია. ეს ჯგუფი სულ  $N$  რაოდენობის კუნძულს შეიცავს, რომლებიც გადანომრილია 0-დან  $(N - 1)$ -მდე.

არსებობს  $M$  რაოდენობის კანოე, რომლებიც გადანომრილია 0-დან  $(M - 1)$ -მდე და ისინი კუნძულებს შორის გადასაადგილებლად გამოიყენება. ყოველი  $i$ -სათვის, სადაც  $0 \leq i \leq M - 1$ ,  $i$ -ური კანოე შეიძლება მიმაგრებული იყოს ან  $U[i]$ , ან  $V[i]$  კუნძულზე და შეიძლება გამოყენებული იქნას  $U[i]$  და  $V[i]$  კუნძულებს შორის გადასაადგილებლად. კერძოდ, როცა კანოე მიმაგრებულია  $U[i]$  კუნძულზე, მისი გამოყენება შესაძლებელია  $U[i]$  კუნძულიდან  $V[i]$  კუნძულზე მისასვლელად, რის შემდეგაც ის  $V[i]$  კუნძულს მიემაგრება. ანალოგიურად, როცა კანოე მიმაგრებულია  $V[i]$  კუნძულზე, მისი გამოყენება შესაძლებელია  $V[i]$  კუნძულიდან  $U[i]$  კუნძულზე მისასვლელად, რის შემდეგაც ის  $U[i]$  კუნძულს მიემაგრება. თავდაპირველად კანოე მიმაგრებულია  $U[i]$  კუნძულზე. კუნძულთა ერთი და იგივე წყვილს შორის გადასაადგილებლად შესაძლებელია რამდენიმე კანოე გამოიყენებოდეს. ასევე შესაძლებელია, რომ ერთი და იგივე კუნძულზე რამდენიმე კანოე იყოს მიმაგრებული.

ცურვის უსაფრთხოების უზრუნველსაყოფად აუცილებელია კანოე შეკეთებული იქნას ყოველი ცურვის შემდეგ, რის გამოც აკრძალულია ერთი და იგივე კანოეთი ორჯერ ზედიზედ ცურვა. ანუ,  $i$ -ური კანოეთი ცურვის შემდეგ, მის ხელახლა გამოყენებამდე აუცილებლად სხვა კანოე უნდა იქნას გამოყენებული.

ბუ დენგკლევს ზოგიერთ კუნძულებზე მოგზაურობის დაგეგმვა სურს. მისი მოგზაურობა **ვალიდურია** (დასაშვებია) მაშინ და მხოლოდ მაშინ, თუ სრულდება შემდეგი პირობები:

- ის მოგზაურობას იწყებს 0-ვანი კუნძულიდან და ამთავრებს ამავე კუნძულზე.
- ის ეწვევა 0-ვანი კუნძულის გარდა კიდევ ერთ კუნძულს მაინც.
- მოგზაურობის დამთავრების შემდეგ თითოეული კანოე მიმაგრებული იქნება იგივე კუნძულზე, რომელზეც იგი მოგზაურობის დაწყების წინ იყო მიმაგრებული. ანუ, ყოველი  $i$ -სათვის, სადაც  $0 \leq i \leq M - 1$ ,  $i$ -ური კანოე  $U[i]$  კუნძულზე უნდა იყოს მიმაგრებული.

დაეხმარეთ ბუ დენგკლევს იპოვოს ნებისმიერი ვალიდური მოგზაურობა, რომელიც კუნძულებს შორის არაუმეტეს 2 000 000 ცურვას შეიცავს, ან დაადგინეთ, რომ არცერთი ასეთი ვალიდური მოგზაურობა არ არსებობს. შეიძლება დამტკიცდეს, რომ იმ შეზღუდვებით, რომლებიც ამოცანაშია მოცემული (იხილეთ შეზღუდვების განყოფილება), თუ ვალიდური მოგზაურობა არსებობს, მაშინ ასევე არსებობს ისეთი ვალიდური მოგზაურობაც, რომელიც კუნძულებს შორის არაუმეტეს 2 000 000 ცურვას შეიცავს.

# იმპლემენტაციის დეტალები

თქვენ უნდა მოახდინოთ შემდეგი პროცედურის იმპლემენტაცია:

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

- $N$ : კუნძულების რაოდენობა.
- $M$ : კანოეთა რაოდენობა.
- $U, V$ :  $M$  სიგრძის მასივები, რომლებითაც კანოეები არის აღწერილი.
- ამ პროცედურამ უნდა დააბრუნოს ან ლოგიკური მნიშვნელობა, ან მთელ რიცხვთა მასივი.
  - თუ ვალიდური მოგზაურობა არ არსებობს, მაშინ პროცედურამ უნდა დააბრუნოს `false`.
  - თუ ვალიდური მოგზაურობა არსებობს, მაშინ თქვენ გაქვთ ორი ვარიანტი:
    - სრული ქულის მისაღებად პროცედურამ უნდა დააბრუნოს მასივი, რომელიც არაუმეტეს 2 000 000 მთელ რიცხვს შეიცავს და რომელიც ვალიდურ მოგზაურობას აღწერს. უფრო ზუსტად, ამ მასივის ელემენტები უნდა წარმოადგენდნენ იმ კანოეების ნომრებს, რომლებიც მოგზაურობისას იქნენ გამოყენებული (მათი გამოყენების რიგის მიხედვით).
    - ნაწილობრივი ქულების მისაღებად, პროცედურამ უნდა დააბრუნოს მნიშვნელობა `true`, 2 000 000-ზე მეტი მთელი რიცხვის შემცველი მასივი, ან მთელ რიცხვთა ისეთი მასივი, რომელიც არ აღწერს ვალიდურ მოგზაურობას. (მეტი დეტალებისათვის იხილეთ ქვეამოცანების განყოფილება).
- ეს პროცედურა გამოიძახება მხოლოდ ერთხელ.

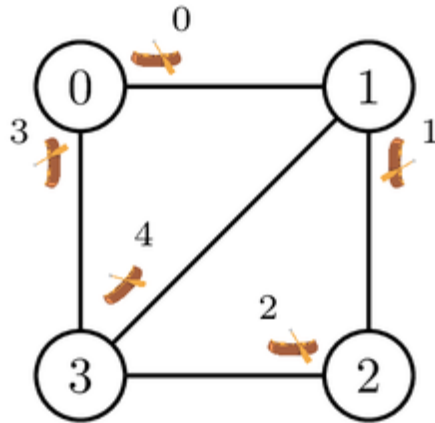
## მაგალითები

### მაგალითი 1

განვიხილოთ შემდეგი გამოძახება:

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

კუნძულები და კანოეები ქვემოთ მოცემულ ნახაზზეა ნაჩვენები.



ერთ-ერთი შესაძლებელი ვალიდური მოგზაურობა არის შემდეგი მოგზაურობა: ბუ დენგკლევი თავდაპირველად გამოიყენებს 0, 1, 2 და 4 კანოებს (ამ მიმდევრობით). შედეგად ის აღმოჩნდება კუნძულზე ნომრით 1. ამის შემდეგ მას ისევ შეუძლია იცუროს კანოეთი 0, რადგან ეს კანოე მოცემულ მომენტში მიმაგრებულია კუნძულზე 1 და ის უშუალოდ წინა ცურვისას არ იყო გამოყენებული. კანოე 0-ის კვლავ გამოყენების შემდეგ ბუ დენგკლევი კვლავ აღმოჩნდება კუნძულზე 0. თუმცა, კანოეები 1, 2 და 4 არ არიან მიმაგრებული იმავე კუნძულებზე, რომლებზეც ისინი მოგზაურობის დასაწყისში იყვნენ მიმაგრებული. ამიტომ, ბუ დენგკლევი აგრძელებს თავის მოგზაურობას და კვლავ იყენებს კანოეებს 3, 2, 1, 4 და 3. საბოლოოდ, იგი ისევ აღმოჩნდება კუნძულზე 0 და ახლა უკვე ყველა კანოე ისევ იგივე კუნძულებზე იქნება მიმაგრებული, რომლებზეც მოგზაურობის წინ.

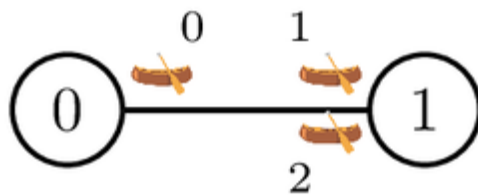
შესაბამისად, დაბრუნებული მნიშვნელობა  $[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]$  ვალიდურ მოგზაურობას წარმოადგენს.

## მაგალითი 2

განვიხილოთ შემდეგი გამოცხება:

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

კუნძულები და კანოეები ქვემოთ მოცემულ ნახაზზეა ნაჩვენები.



ბუ დენგკლეკს მოგზაურობის დაწყება შეუძლია მხოლოდ კანოეთი 0, ამის შემდეგ მას შეუძლია იცუროს ან კანოეთი 1, ან კანოეთი 2 (შევნიშნოთ, რომ იგი ვერ გამოიყენებს კანოეს 0 ზედიზედ ორჯერ). ორივე შემთხვევაში ბუ დენგკლეკი ისევ აღმოჩნდება კუნძულზე 0. თუმცა, კანოეები არ არიან მიმაგრებული იმავე კუნძულებზე, რომლებზეც ისინი მოგზაურობის დასაწყისში იყვნენ მიმაგრებული და ამის შემდეგ მას არცერთი კანოეს გამოყენება აღარ შეუძლია, რადგან ერთადერთი კანოე, რომელიც კუნძულზე 0 არის მიმაგრებული, არის ის კანოე, რომელიც მან ახლახანს გამოიყენა. ამიტომ, რადგანაც ვალიდური მოგზაურობა ამ შემთხვევაში არ არსებობს, პროცედურამ უნდა დააბრუნოს false.

## შეზღუდვები

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 200\,000$
- $0 \leq U[i] \leq N - 1$  და  $0 \leq V[i] \leq N - 1$  (თითოეული  $i$ -სათვის, სადაც  $0 \leq i \leq M - 1$ )
- $U[i] \neq V[i]$  (თითოეული  $i$ -სათვის, სადაც  $0 \leq i \leq M - 1$ )

## ქვეამოცანები

1. (5 ქულა)  $N = 2$
2. (5 ქულა)  $N \leq 400$ .  $x$  და  $y$  ( $0 \leq x < y \leq N - 1$ ) განსხვავებული კუნძულების თითოეული წყვილისათვის არსებობს ზუსტად ორი კანოე, რომლებიც მათ შორის გადასაადგილებლად შეიძლება იქნენ გამოყენებული. ერთ-ერთი მათგანი მიმაგრებულია კუნძულზე  $x$ , ხოლო მეორე კი - კუნძულზე  $y$ .
3. (21 ქულა)  $N \leq 1000$ ,  $M$  ლუწია და ყოველი ლუწი  $i$ -სათვის, სადაც  $0 \leq i \leq M - 1$ , კანოეები  $i$  და  $i + 1$  ორივე შეიძლება გამოყენებული იქნას  $U[i]$  და  $V[i]$  კუნძულებს შორის გადასაადგილებლად. კანოე  $i$  თავდაპირველად მიმაგრებულია კუნძულზე  $U[i]$ , ხოლო კანოე  $i + 1$  კი - კუნძულზე  $V[i]$ . ფორმალურად,  $U[i] = V[i + 1]$  და  $V[i] = U[i + 1]$ .
4. (24 ქულა)  $N \leq 1000$ ,  $M$  ლუწია და ყოველი ლუწი  $i$ -სათვის, სადაც  $0 \leq i \leq M - 1$ , კანოეები  $i$  და  $i + 1$  ორივე შეიძლება გამოყენებული იქნას  $U[i]$  და  $V[i]$  კუნძულებს შორის გადასაადგილებლად. ორივე კანოე თავდაპირველად მიმაგრებულია კუნძულზე  $U[i]$ . ფორმალურად,  $U[i] = U[i + 1]$  and  $V[i] = V[i + 1]$ .
5. (45 ქულა) დამატებითი შეზღუდვების გარეშე.

თითოეული ტესტისათვის, რომელშიც ვალიდური მოგზაურობა არსებობს, თქვენი ამოხსნა:

- მიიღებს სრულ ქულას, თუ ის აბრუნებს ვალიდურ მოგზაურობას,
- მიიღებს ქულების 35%-ს, თუ აბრუნებს მნიშვნელობას true, მასივს 2 000 000-ზე მეტი მთელი რიცხვით, ან მასივს, რომელიც არ აღწერს ვალიდურ მოგზაურობას.
- მიიღებს 0 ქულას წინააღმდეგ შემთხვევაში.

თითოეული ტესტისათვის, რომელშიც ვალიდური მოგზაურობა არ არსებობს, თქვენი ამოხსნა:

- მიიღებს სრულ ქულას, თუ ის აბრუნებს მნიშვნელობას false,
- მიიღებს 0 ქულას წინააღმდეგ შემთხვევაში.

გაითვალისწინეთ, რომ საბოლოო ქულა თითოეული ქვეამოცანისათვის არის ამ ქვეამოცანის ტესტებში მიღებულ ქულათა შორის მინიმალური.

## სანიმუშო გრადერი

სანიმუშო გრადერი შემოსატან მონაცემებს კითხულობს შემდეგი ფორმატით:

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

სანიმუშო გრადერი ბეჭდავს თქვენს პასუხებს შემდეგი ფორმატით:

- თუ `find_journey` აბრუნებს `bool`:
  - სტრიქონი 1: 0
  - სტრიქონი 2: 0 თუ `find_journey` აბრუნებს `false`, ან 1 წინააღმდეგ შემთხვევაში.
- თუ `find_journey` აბრუნებს `int[]`, აღვნიშნოთ ამ მასივის ელემენტები  $c[0], c[1], \dots, c[k - 1]$ .  
სანიმუშო გრადერი ბეჭდავს:
  - სტრიქონი 1: 1
  - სტრიქონი 2:  $k$
  - სტრიქონი 3:  $c[0] \ c[1] \ \dots \ c[k - 1]$