



## Minlər Adaları

Minlər Adaları Java dənizində yerləşən gözəl adalar qrupudur. Bu adalar qrupu 0-dan  $N - 1$ -ə nömrələnmiş  $N$  sayda adadan ibarətdir.

Orada həmçinin 0-dan  $M - 1$ -ə nömrələnmiş  $M$  sayda kanoe var və onlardan istifadə edib adalar arasında səyahət edə bilərsiniz.  $i$ -ci ( $0 \leq i \leq M - 1$ ) kanoe  $U[i]$  və ya  $V[i]$  adasında dayana bilər. Onlardan istifadə edərək  $U[i]$  və  $V[i]$  adaları arasında səyahət edə bilərsiniz. Daha dəqiq desək,  $U[i]$  adasında dayanmış kanoe ilə  $U[i]$  adasından  $V[i]$  adasına gedə bilərsiniz, bu səyahətdən sonra kanoe  $V[i]$  adasında dayanmış olacaq. Oxşar olaraq,  $V[i]$  adasında dayanmış kanoe ilə  $V[i]$  adasından  $U[i]$  adasına gedə bilərsiniz, bu səyahətdən sonra kanoe  $U[i]$  adasında dayanmış olacaq. Başda, kanoelər  $U[i]$  adasında dayanıblar. Ola bilər ki, iki ada arasında çoxlu sayda qayıq səyahət etsin. Həmçinin, eyni adada bir neçə kanoe dayana bilər.

Təhlükəsizlik səbəblərindən, hər bir kanoe səyahət etdikdən sonra baxışdan keçirilməlidir. Buna görə də, eyni qayıqla iki dəfə arxa-arkaya səyahət edə bilməzsiniz. Yəni,  $i$ -ci qayıq istifadə etdikdən sonra, ondan bir daha istifadə etmək üçün əvvəlcə başqa kanoedən istifadə etməlisiniz.

Adilə bu adalar arasında səfər planlayır. Əgər aşağıdakı şərtlər ödənərsə, o zaman onun səfəri **düzgün** sayılır.

- Onun səfəri 0-cı adada başlayır və bitir.
- 0-cı adadan başqa ən az bir adaya gedir.
- Səfər bitdikdən sonra hər bir kanoe səfərin başında olduqları yerdə dayanmış olur. Yəni, kanoe  $i$  ( $0 \leq i \leq M - 1$ )  $U[i]$  adasında dayanmış olur.

Adiləyə ən çox 2 000 000 səyahət etməklə düzgün səfər tapmağa kömək edin, ya da belə bir səfərin mümkün olmadığını bildirin. İsbat etmək olar ki, verilmiş məhdudiyətlər daxilində (Məhdudiyətlər bölməsinə baxın) əgər düzgün səfər varsa, onda 2 000 000-dan çox olmayan sayda səyahətdən ibarət düzgün səfər də var.

## İmplementasiya detalları

Siz aşağıdakı proseduru implement etməlisiniz:

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

- $N$ : adaların sayı.

- $M$ : kanoelərin sayı.
- $U, V$ : kanoeləri təsvir edən  $M$  uzunluqlu massivlər.
- Bu prosedur ya boolean dəyər, ya da tam ədədlər massivi qaytarmalıdır.
  - Əgər düzgün səfər yoxdursa false qaytarmalıdır.
  - Əgər düzgün səfər varsa iki seçiminiz var:
    - Tam bal almaq üçün prosedur səfəri təsvir edən, 2 000 000-dən uzun olmayan tam ədədlər massivi qaytarmalıdır. Daha dəqiq olsaq, qaytarılan massivin elementləri səfər zamanı istifadə olunan kanoelərin nömrələrinə bərabər olmalıdır (istifadə olunduqları sırada).
    - Yarımçıq bal almaq üçün prosedur true, 2 000 000-dən uzun massiv, və ya düzgün olmayan səfər qaytarmalıdır (Daha detallı məlumat üçün alt tapşırıqlar bölməsinə baxın).
- Bu prosedur bir dəfə çağırılır.

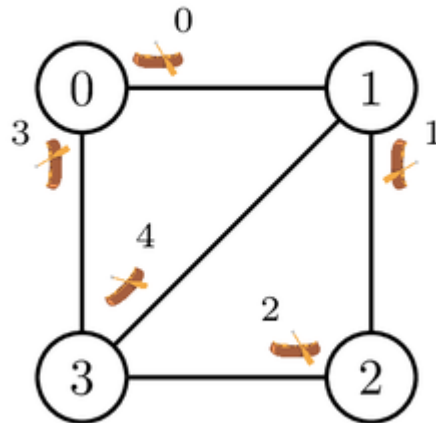
## Nümunələr

### Nümunə 1

Aşağıdakı prosedura nəzər yetirin:

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

Adalar və kanoelər aşağıdakı şəkildə göstərilib.



Mümkün səfərlərdən biri belədir. Adilə əvvəlcə sıra ilə 0, 1, 2, və 4 kanoeləri ilə səyahət edir. Bundan sonra o, 1 adasında olacaq. Son istifadə etdiyi kanoenin nömrəsi 0 olmadığından, həmin kanoeni yenidən istifadə edə bilər. Beləcə, hazırda 1 nömrəli adada dayanmış 0 nömrəli kanoə ilə yenidən səyahət edib 0 nömrəli adaya qayıda bilər. Lakin, 1, 2, və 4 nömrəli kanoelər başda olduqları yerdə dayanmayıblar. Buna görə də Adilənin səfəri 3, 2, 1, 4, və 3 nömrəli kanoelərlə davam edir. Bundan sonra bütün kanoelər başda olduqları yerdədilər və Adilə 0 nömrəli adadadır.

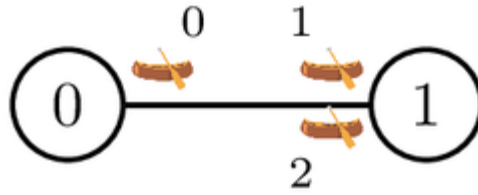
Buna görə də prosedur  $[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]$  massivini qaytara bilər.

## Nümunə 2

Aşağıdakı prosedura nəzər yetirin:

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

Adalar və kanoelər aşağıdakı şəkildə göstərilib.



Adilə ancaq 0 nömrəli kanoedən istifadə edərək səfərinə başlaya bilər. Son səyahəti 0 nömrəli kanoelə olduqundan, yenidən onunla səyahət edə bilməz. Bundan sonra o ancaq 1 və ya 2 nömrəli kanoelərdən istifadə edə bilər. Hər iki halda yenidən 0 nömrəli adaya qayıdır. Bütün kanoelər başladıkları yerdə olmadıqları üçün səyahət bitməyib. Lakin, adada yenidən geri qayıtmaq üçün olan yeganə kanoedən son səyahət üçün istifadə olunub və yenidən onunla səyahət etmək olmaz. Heç bir düzgün səfər olmadığı üçün, prosedur false qaytarmalıdır.

## Məhdudiyyətlər

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

## Alt tapşırıqlar

1. (5 bal)  $N = 2$
2. (5 bal)  $N \leq 400$ . Bütün müxtəlif  $x$  və  $y$  adaları üçün ( $0 \leq x < y \leq N - 1$ ), aralarında səyahət etməyə tam olaraq 2 kanoelə var. Onlardan biri  $x$ , digəri isə  $y$  adasında dayanıblar.
3. (21 bal)  $N \leq 1000$ ,  $M$  cüt ədəddir, və hər **cüt**  $i$  üçün ( $0 \leq i \leq M - 1$ ),  $i$  və  $i + 1$  nömrəli kanoelərin hər biri  $U[i]$  və  $V[i]$  adaları arasında səyahət etmək üçün istifadə oluna bilər. Kanoelə  $i$  başda  $U[i]$  adasında, kanoelə  $i + 1$  isə  $V[i]$  adasında dayanıb. Formal olaraq yazsaq,  $U[i] = V[i + 1]$  və  $V[i] = U[i + 1]$ .

4. (24 bal)  $N \leq 1000$ ,  $M$  cütdür, və bütün **cüt**  $i$  üçün ( $0 \leq i \leq M - 1$ ),  $i$  və  $i + 1$  nömrəli kanoelərin hər biri  $U[i]$  və  $V[i]$  adaları arasında səyahət etmək üçün istifadə oluna bilər. Hər iki kanoe başda  $U[i]$  adasında dayanıb. Formal olaraq yazsaq,  $U[i] = U[i + 1]$  və  $V[i] = V[i + 1]$ .

5. (45 bal) Əlavə məhdudiyətlər yoxdur.

Düzgün səfərin olduğu hər testdə sizin həllinizin alacağı ballar aşağıdakı kimidir:

- əgər düzgün həlldirsə – tam bal,
- əgər prosedur `true`, 2 000 000-dan uzun massiv, və ya düzgün olmayan səfəri qaytarsa – mümkün balın 35%-i
- digər hallarda – 0 bal

Düzgün səfər olmayan hər test üçün sizin həllinizin alacağı ballar aşağıdakı kimidir:

- əgər `false` qaytarsa – tam bal
- digər hallarda – 0 bal

Qeyd edək ki, hər bir alt tapşırığın balı, həmin alt tapşırıqdakı minimum bal almış testin balına bərabər olacaq.

## Nümunə qreyder

Nümunə qreyder giriş verilənlərini aşağıdakı formatda oxuyur:

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

Nümunə qreyder sizin cavablarınızı aşağıdakı formatda çap edir:

- Əgər `find_journey` proseduru `bool` dəyər qaytarsa:
  - sətir 1: 0
  - sətir 2: əgər `find_journey` proseduru `false` qaytarsa 0, əks halda 1.
- Əgər `find_journey` proseduru `int[]` qaytarsa, bu massivin elementlərini  $c[0], c[1], \dots, c[k - 1]$  ilə işarələyin. Nümunə qreyder çıxışa bunları verir:
  - sətir 1: 1
  - sətir 2:  $k$
  - sətir 3:  $c[0] c[1] \dots c[k - 1]$