



## Tūkstošiem salu

Tūkstošiem salu ir skaistu salu grupa, kas atrodas Javas jūrā. Tā sastāv no  $N$  salām, kas numurētas no 0 līdz  $N - 1$ .

Ir  $M$  kanoe laivas, kas sanumurētas no 0 līdz  $M - 1$ , ar kurām var kuģot starp salām. Katram  $i$ , kur  $0 \leq i \leq M - 1$ , kanoe laiva  $i$  var piestāt vai nu salā  $U[i]$ , vai  $V[i]$ , un to var izmantot, lai kuģotu starp salām  $U[i]$  un  $V[i]$ . Konkrēti, kad kanoe laiva ir pietauvota salā  $U[i]$ , to var izmantot, lai kuģotu no salas  $U[i]$  uz salu  $V[i]$ , pēc tam kanoe laiva tiek pietauvota salā  $V[i]$ . Līdzīgi, kad kanoe laiva ir pietauvota salā  $V[i]$ , to var izmantot, lai kuģotu no salas  $V[i]$  uz salu  $U[i]$ , pēc tam kanoe laiva tiek pietauvota salā  $U[i]$ . Sākotnēji kanoe laiva ir pietauvota salā  $U[i]$ . Iespējams, ka starp vienu un to pašu salu pāri var izmantot vairākas kanoe laivas. Iespējams arī, ka vienā salā ir pietauvotas vairākas kanoe laivas.

Drošības apsvērumu dēļ kanoe laiva ir jāapkopj pēc katra brauciena, kas aizliedz vienu un to pašu kanoe laivu divas reizes pēc kārtas. Tas nozīmē, ka pēc kādas kanoe laivas  $i$  izmantošanas ir jāizmanto cita kanoe laiva, lai varētu atkal izmantot kanoe laivu  $i$ .

Bū Dengkleka vēlas izplānot ceļojumu pa dažām no salām. Viņas ceļojums ir **derīgs** tad un tikai tad, ja ir izpildīti šādi nosacījumi:

- Viņa sāk un beidz savu ceļojumu salā 0.
- Viņa apmeklē vismaz vienu salu, neskaitot salu 0.
- Pēc ceļojuma beigām katra kanoe laiva tiek pietauvota tajā pašā salā, kur tā bija pietauvota pirms ceļojuma. T.i., kanoe laivai  $i$ , katram  $i$ , kur  $0 \leq i \leq M - 1$ , jābūt pietauvotai salā  $U[i]$ .

Palīdziet Bū atrast jebkuru derīgu ceļojumu, kas ietver kuģošanu ne vairāk kā 2 000 000 reizes, vai arī noteikt, ka šāds ceļojums nepastāv. Var pierādīt, ka, saskaņā ar šajā uzdevumā norādītajiem ierobežojumiem (skat. sadaļu *Ierobežojumi*), ja pastāv derīgs ceļojums, pastāv arī derīgs ceļojums, kas ietver ne vairāk kā 2 000 000 kuģošanas reizes.

## Realizācijas detaļas

Ir jārealizē šāda procedūra:

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

- $N$ : salu skaits.

- $M$ : kanoe laivu skaits.
- $U, V$ : masīvi garumā  $M$ , kas apraksta kanoe laivas.
- Procedūrai jāatgriež vai nu loģiskā vērtība, vai arī veselu skaitļu masīvs.
  - Ja derīgs ceļojums neeksistē, procedūrai jāatgriež vērtība `false`.
  - Ja derīgs ceļojums eksistē, jums ir divas iespējas:
    - Lai saņemtu maksimālo punktu skaitu, procedūrai jāatgriež masīvs ar ne vairāk kā 2 000 000 veseliem skaitļiem, kas apraksta derīgu ceļojumu. Precīzāk, masīva elementiem jābūt kanoe laivu numuriem tieši tādā secībā, kādā tās tiek izmantotas ceļojuma laikā.
    - Lai saņemtu daļēju punktu skaitu, procedūrai jāatgriež vērtība `true`, masīvs ar vairāk nekā 2 000 000 veseliem skaitļiem vai veselu skaitļu masīvs, kas neapraksta derīgu ceļojumu (detalizētu aprakstu skat. sadaļā *Apakšuzdevumi*).
- Šī procedūra tiek izsaukta tieši vienreiz.

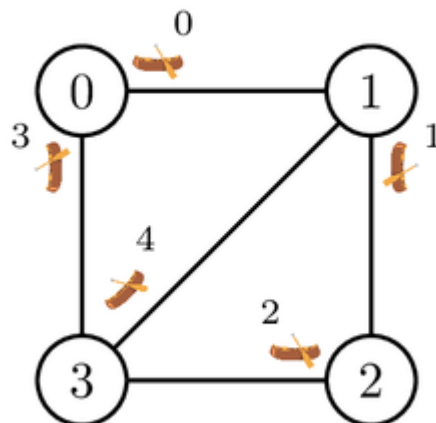
## Piemēri

### 1. piemērs

Aplūkosim šādu izsaukumu:

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

Salas un kanoe laivas ir parādītas zīmējumā:



Aprakstīsim vienu derīgu ceļojumu. Bū vispirms kuģo ar kanoe laivām 0, 1, 2 un 4, tieši šādā secībā. Rezultātā viņa atrodas salā 1. Pēc tam Bū var atkal kuģot ar kanoe 0, jo tā šobrīd ir pietauvota salā 1, un pēdējā viņas izmantotā kanoe laiva nav 0. Pēc kuģošanas ar kanoe laivu 0, Bū nonāk salā 0. Diemžēl, kanoe laivas 1, 2 un 4 neatrodas tajās salās, kur tās bija pirms ceļojuma. Tāpēc Bū turpina savu ceļojumu, kuģojot ar kanoe laivām 3, 2, 1, 4 un atkal ar 3. Bū ir atgriezies salā 0 un visas kanoe laivas ir pietauvotas tajās salās, kurās tās atradās pirms ceļojuma.

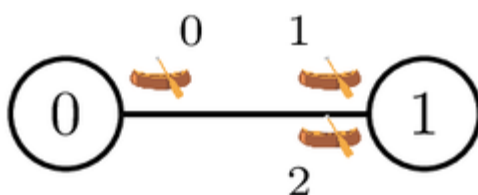
Tādējādi, atgriežamā vērtība  $[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]$  apraksta derīgu ceļojumu.

## 2. piemērs

Aplūkosim šādu izsaukumu:

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

Salas un kanoe laivas ir parādītas zīmējumā:



Bū ceļojumu var sākt vienīgi kuģošanai izmantojot kanoe laivu 0, bet pēc tam izmantot vai nu kanoe laivu 1, vai 2. Ievērojiet, ka viņa nevar kuģot divreiz pēc kārtas ar kanoe laivu 0. Abos gadījumos Bū būs atgriezies salā 0. Tomēr kanoe laivas neatrodas tajās salās, kurās tās atradās pirms ceļojuma, un Bū nevar turpināt ceļojumu, jo vienīgā laiva, kas tagad atrodas salā 0, ir tā kuru viņa tikko izmantoja. Tā kā derīga ceļojuma nav, procedūrai jāatgriež vērtība false.

## Ierobežojumi

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

## Apakšuzdevumi

1. (5 punkti)  $N = 2$
2. (5 punkti)  $N \leq 400$ . Katram atšķirīgu salu pārim  $x$  un  $y$  ( $0 \leq x < y \leq N - 1$ ), ir tieši divas kanoe laivas, kas var tikt izmantotas kuģojumam starp tām. Viena no kanoe laivām ir pietauvota salā  $x$ , bet otra – salā  $y$ .
3. (21 punkts)  $N \leq 1000$ ,  $M$  ir pāru skaitlis, un katram **pāru** skaitlim  $i$ , kur  $0 \leq i \leq M - 1$ , kanoe laivas  $i$  un  $i + 1$  abas var izmantot kuģošanai starp salām  $U[i]$  un  $V[i]$ . Kanoe laiva  $i$  sākotnēji ir pietauvota salā  $U[i]$ , un kanoe laiva  $i + 1$  sākotnēji ir pietauvota salā  $V[i]$ . Formāli,  $U[i] = V[i + 1]$  un  $V[i] = U[i + 1]$ .

4. (24 punkti)  $N \leq 1000$ ,  $M$  ir pāru skaitlis, un katram **pāru** skaitlim  $i$ , kur  $0 \leq i \leq M - 1$ , kanoē laivas  $i$  un  $i + 1$  abas var izmantot kuģošanai starp salām  $U[i]$  un  $V[i]$ . Abas kanoē laivas sākotnēji ir pietauvotas salā  $U[i]$ . Formāli,  $U[i] = U[i + 1]$  un  $V[i] = V[i + 1]$ .
5. (45 punkti) Bez papildu ierobežojumiem.

Katram testam, kur eksistē derīgs ceļojums, jūsu risinājums saņems:

- maksimālo punktu skaitu, ja atgriezīs derīgu ceļojumu,
- 35% no maksimālā punktu skaita, ja atgriezīs vērtību `true`, masīvu ar vairāk nekā 2 000 000 veseliem skaitļiem, vai masīvu, kas neapraksta derīgu ceļojumu,
- 0 punktus visos citos gadījumos.

Katram testam, kur derīgs ceļojums neeksistē, jūsu risinājums saņems:

- maksimālo punktu skaitu, ja atgriezīs vērtību `false`,
- 0 punktus pretējā gadījumā.

Ievērojiet, ka galīgais punktu skaits par apakšuzdevumu ir mazākais punktu skaits starp šī apakšuzdevuma testiem.

## Paraugvērtētājs

Paraugvērtētājs lasa ievaddatus šādā formātā:

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

Paraugvērtētājs izdrukā jūsu atbildes šādā formātā:

- Ja `find_journey` izvada `bool` tipa atbildi:
  - 1. rinda: 0
  - 2. rinda: 0, ja `find_journey` atgriež `false`, vai 1 pretējā gadījumā.
- Ja `find_journey` atgriež `int[]` tipa atbildi, apzīmēsim šī masīva elementus attiecīgi kā  $c[0], c[1], \dots, c[k - 1]$ . Paraugvērtētājs izdrukās:
  - 1. rinda: 1
  - 2. rinda:  $k$
  - 3. rinda:  $c[0] c[1] \dots c[k - 1]$