



Tuhanded saared

Tuhanded saared on ilus arhipelaag Java meres. See koosneb N saarest, mis on nummerdatud 0 kuni $N - 1$.

Saarte vahel sõitmiseks saab kasutada M kanuud, mis on nummerdatud 0 kuni $M - 1$. Iga i korral, kus $0 \leq i \leq M - 1$, saab kanuu i randuda kas saarel $U[i]$ või $V[i]$ ning sellega saab sõita saarte $U[i]$ ja $V[i]$ vahel. Täpsemalt, kui kanuu on saare $U[i]$ rannas, saab seda kasutada sõitmiseks saarelt $U[i]$ saarele $V[i]$. Seejärel asub kanuu saare $V[i]$ rannas. Samamoodi, kui kanuu asub saare $V[i]$ rannas, saab seda kasutada sõitmiseks saarelt $V[i]$ saarele $U[i]$. Seejärel asub kanuu saare $U[i]$ rannas. Alguses on kanuu saare $U[i]$ rannas. On võimalik, et sama saartepaari vahel saab sõita mitme erineva kanuuga. Samuti on võimalik, et ühe saare rannas on mitu kanuud.

Ohutuse mõttes peab iga kanuu pärast sõitu läbima hoolduse, seetõttu ei saa sama kanuuga kaks korda järjest sõita. See tähendab, et pärast mingi kanuuga i sõitmist peab vahepeal kasutama mõnd teist kanuud, enne kui jälle kanuud i saab kasutada.

Bu Dengklek plaanib reisida läbi mõnede saarte. Tema reis on **korralik** parajasti siis, kui on täidetud järgmised tingimused:

- Reis algab ja lõpeb saarel 0.
- Ta käib vähemalt ühel saarel, mis ei ole saar 0.
- Pärast reisi lõppu on kõik kanuud samal saarel, kus nad enne reisi asusid. S.t kanuu i , iga i korral, kus $0 \leq i \leq M - 1$, peab olema saare $U[i]$ rannas.

Aita Bu Dengklekil leida mõni korralik reis, mis ei vajaks rohkem kui 2 000 000 üksikut sõitu, või ütle, et sellist reisi pole võimalik korraldada. Saab tõestada, et kui ülesande tingimustes antud piirangute puhul (vt Piirangud) leidub mõni korralik reis, siis leidub ka korralik reis, mis koosneb mitte rohkem kui 2 000 000 sõidust.

Realisatsioon

Sul tuleb kirjutada järgmine funktsioon:

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

- N : saarte arv.
- M : kanuude arv.

- U, V : massiivid pikkusega M , mis kirjeldavad kanuude marsruute.
- Funktsioon peab tagastama kas tõeväärtuse või täisarvude massiivi.
 - Kui korralikku reisi ei leidu, peab funktsioon tagastama `false`.
 - Kui korralik reis on võimalik, on sul kaks võimalust:
 - Et saada täispunkte, peab funktsioon tagastama täisarvude massiivi, mille pikkus on ülimalt 2 000 000, mis kirjeldab korralikku reisi. Täpsemalt peavad massiivi elemendid vastama reisi jaoks kasutatavate kanuude numbritele (kasutamise järjekorras).
 - Et saada osalisi punkte, peab funktsioon tagastama `true`, massiivi, kus on rohkem kui 2 000 000 täisarvu või massiivi, milles olevad täisarvud ei kirjelda korralikku reisi. (Vaata lähemalt seksiooni Alamülesanded).
- Funktsiooni kutsutakse välja täpselt üks kord.

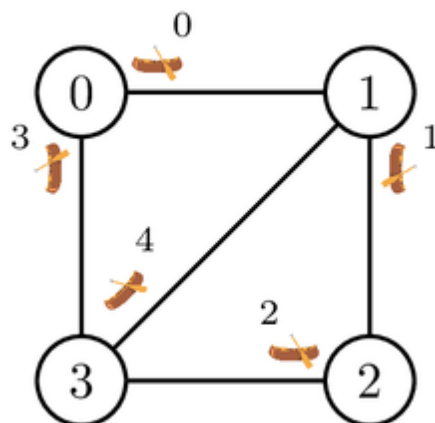
Näited

Näide 1

Vaatleme järgmist väljakutset:

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

Kirjeldatud saared ja kanuud on näidatud järgneval pildil.



Üks võimalik reis on järgmine: Bu Dengklek sõidab kõigepealt kanuudega 0, 1, 2 ja 4, parajasti selles järjekorras. Tulemusena on ta saarel 1. Seejärel saab Bu Dengklek jälle kanuud 0 kasutada, kuna see on saare 1 rannas ning viimane kanuu, mida ta kasutas, ei ole kanuu 0. Pärast uuesti kanuuga 0 sõitmist on Bu Dengklek nüüd saarel 0. Kanuud 1, 2 ja 4 ei ole aga samadel saartel, kus nad reisi alguses olid. Bu Dengklek jätkab oma teekonda ning sõidab uuesti kanuudega 3, 2, 1, 4 ja 3. Bu Dengklek on nüüd tagasi saarel 0 ja kõik kanuud on samadel saartel, kus reisi alguseski.

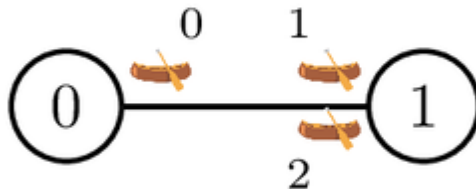
Seega esitab `[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]` korralikku reisi.

Näide 2

Vaatleme järgmist väljakutset:

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

Kirjeldatud saared ja kanuud on näidatud järgneval pildil.



Bu Dengklek saab reisi alustada ainult kanuuga 0, mille järel ta võib võtta kas kanuu 1 või 2. Pane tähele, et kanuuga 0 ei saa kaks korda järjest sõita. Mõlemal juhul on Bu Dengklek tagasi saarel 0. Nüüd ei ole kanuud aga samadel saartel, kus nad alguses olid, ja Bu Dengklek ei saa enam ühegi kanuuga sõita, sest ainus kanuu saarel 0 on see, mida ta just kasutas. Kuna korralikku reisi ei ole, peab funktsioon tagastama false.

Piirangud

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

Alamülesanded

1. (5 punkti) $N = 2$
2. (5 punkti) $N \leq 400$. Iga erinevate saarte paari x ja y ($0 \leq x < y \leq N - 1$) jaoks on täpselt kaks kanuud, millega nende vahel saab sõita. Üks neist on saarel x ja teine on saarel y .
3. (21 punkti) $N \leq 1000$, M on paaris ja iga **paarisarvulise** i korral, kus $0 \leq i \leq M - 1$, saab kasutada kanuusid i ja $i + 1$, et sõita saarte $U[i]$ ja $V[i]$ vahel. Kanuu i on alguses saare $U[i]$ rannas ja kanuu $i + 1$ on alguses saare $V[i]$ rannas. Formaalset $U[i] = V[i + 1]$ ja $V[i] = U[i + 1]$.
4. (24 punkti) $N \leq 1000$, M on paaris ja iga **paarisarvulise** i korral, kus $0 \leq i \leq M - 1$, saab kasutada kanuusid i ja $i + 1$, et sõita saarte $U[i]$ ja $V[i]$ vahel. Mõlemad kanuud on alguses saarel $U[i]$. Formaalset $U[i] = U[i + 1]$ ja $V[i] = V[i + 1]$.
5. (45 punkti) Lisapiirangud puuduvad.

Iga testi puhul, kus korralik reis on võimalik, saab lahendus:

- täispunktid, kui see tagastab nõuetele vastava reisi,
- 35% punktidest, kui funktsioon tagastab `true`, tagastatud massiivis on rohkem kui 2 000 000 täisarvu või see ei kirjelda korralikku reisi,
- 0 punkti muudel juhtudel.

Iga testi puhul, kus korralik reis ei ole võimalik, saab lahendus:

- täispunktid, kui see tagastab `false`,
- 0 punkti muudel juhtudel.

Pane tähele, et iga alamülesande lõplik skoor on kõigi testide skooride miinimum.

Näidishindaja

Näidishindaja loeb sisendit järgmises vormingus:

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

Näidishindaja kirjutab sinu vastuse järgmises vormingus:

- Kui `find_journey` tagastab `bool` tüüpi väärtuse:
 - rida 1: 0
 - rida 2: 0 kui `find_journey` tagastab `false`, 1 vastasel korral.
- Kui `find_journey` tagastab `int[]`, tähistame massiivi elemendid $c[0], c[1], \dots, c[k - 1]$.

Näidishindaja kirjutab:

- rida 1: 1
- rida 2: k
- rida 3: $c[0] c[1] \dots c[k - 1]$