



Thousands Islands

Thousands Islands skup je predivnih otoka koji se nalaze u moru Java. Skup ukupno broji N otoka savršenih za medeni mjesec, označenih cijelim brojevima od 0 do $N - 1$.

Na otocima se ukupno nalazi i M kanua, označenih cijelim brojevima od 0 do $M - 1$, kojim je moguće putovati između otoka. Za svaki i takav da je $0 \leq i \leq M - 1$, i -ti kanu može se usidriti ili na otoku $U[i]$ ili na otoku $V[i]$, i može se koristiti za putovanje između otoka $U[i]$ i $V[i]$. Preciznije, kada je kanu usidren na otoku $U[i]$, može se koristiti za putovanje od otoka $U[i]$ ka otoku $V[i]$, nakon čega postaje usidren na otoku $V[i]$. Analogno, kada je kanu usidren na otoku $V[i]$, može se koristiti za putovanje od otoka $V[i]$ ka otoku $U[i]$, nakon čega postaje usidren na otoku $U[i]$. Moguće je da postoji više kanua koji se mogu koristiti za putovanje između istog para otoka. Također je moguće je više kanua usidreno u isto vrijeme na nekom otoku.

Zbog sigurnosnih razloga, kanu treba servisirati nakon svakog putovanja, zbog čega se neki kanu ne smije koristiti za dva uzastopna putovanja. Odnosno, nakon korištenja i -tog kanua, potrebno je iskoristiti za putovanje neki drugi kanu prije nego što se ponovno može koristiti kanu i .

Marina i Martin planiraju svoje prvo zajedničko putovanje na kojem će posjetiti neke od otoka. Marina bi radije sve proputovala jahtom, no Martinu su putovanja jahtom već dosadila pa je odlučio upitati svoju prijateljicu Bu Dengklek da im isplanira romantično putovanje.

Bu Dengklek tvrdi da je putovanje **romantično** ako i samo ako su **zadovoljeni** sljedeći uvjeti.

- Putovanje počinje i završava na otoku 0.
- Tijekom putovanja posjećen je barem jedan otok koji nije 0.
- Nakon što putovanje završi, svaki kanu je usidren na istom otoku na kojem je bio i prije nego što je putovanje započelo. Odnosno, i -ti kanu, za svaki i takav da je $0 \leq i \leq M - 1$, mora na kraju putovanja biti usidren na otoku $U[i]$.

Pomozite Bu Dengklek pronaći bilo koje **romantično** putovanje u kojem neće biti više od 2 000 000 putovanja kanuom, ili odredite da takvo putovanje ne postoji. Moguće je dokazati da unutar ograničenja ovog zadatka (vidi poglavlje o ograničenjima) ako **romantično** putovanje postoji, tada također postoji **romantično** putovanje koje ne uključuje više od 2 000 000 putovanja kanuom.

Implementacijski detalji

Potrebno je implementirati sljedeću funkciju:

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

- N : broj otoka.
- M : broj kanua.
- U, V : polja duljine M koja opisuju kanue.
- Funkcija treba vratiti ili `bool` vrijednost ili polje cijelih brojeva.
 - ako romantično putovanje ne postoji, funkcija treba vratiti `false`.
 - ako romantično putovanje postoji, funkcija treba vratiti polje od najviše 2 000 000 cijelih brojeva koji opisuju romantično putovanje. Da biste osvojili sve bodove, elementi ovog polja moraju odgovarati oznakama kanua koji će biti korišteni tijekom putovanja, onim redom kojim će biti korišteni. U protivnom, ako funkcija vrati `true`, ili polje sadrži više od 2 000 000 elemenata, ili polje ne opisuje romantično putovanje, rješenje može osvojiti parcijalne bodove (vidi poglavlje o podzadacima).
- Funkcija će biti pozvana točno jednom.

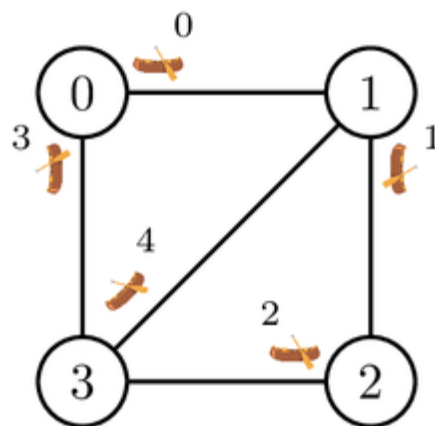
Primjeri

Primjer 1

Razmotrimo sljedeći poziv:

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

Otoci i kanui prikazani su na donjoj slici.



Slijedi opis jednog romantičnog putovanja. Najprije koristimo kanue 0, 1, 2 i 4. Nakon toga, nalazimo se na otoku 1. Potom ponovno koristimo kanu 0 koji se trenutno nalazi usidren na otoku 1 (smijemo jer prethodni kanu kojeg smo koristili nije 0). Nakon putovanja kanuom 0, nalazimo se na otoku 0. Međutim, kanui 1, 2 i 4 nisu usidreni na onim otocima gdje su bili usidreni prije putovanja.

Nastavljamo putovanje korištenjem kanua 3, 2, 1, 4 i 3. Nakon toga se nalazimo na otoku 0 i svi kanui usidreni su na odgovarajućim otocima.

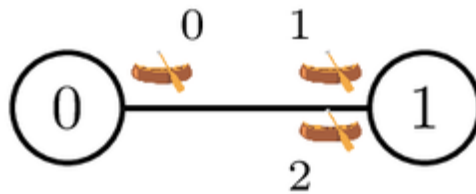
Stoga, $[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]$ je valjana povratna vrijednost funkcije.

Example 2

Razmotrimo sljedeći poziv:

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

Otoci i kanui prikazani su na donjoj slici.



Putovanje možemo započeti korištenjem kanua 0, nakon čega možemo koristiti ili kanu 1 ili kanu 2. Primijetite da ne smijemo koristiti kanu 0 dva puta zaredom. U oba slučaja, ponovno ćemo se naći na otoku 0. Međutim, kanui nisu usidreni na otocima gdje su bili prije putovanja, i ne možemo više iskoristiti niti jedan kanu, budući da je jedini kanu koji se nalazi usidren na otoku 0 upravo onaj koji smo netom prije iskoristili. Budući da ne postoji validno romantično putovanje, funkcija treba vratiti `false`.

Ograničenja

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 200\,000$
- $0 \leq U[i] \leq N - 1$ i $0 \leq V[i] \leq N - 1$ (za svaki i takav da je $0 \leq i \leq M - 1$)
- $U[i] \neq V[i]$ (za svaki i takav da je $0 \leq i \leq M - 1$)

Podzadaci

1. (5 bodova) $N = 2$
2. (5 bodova) $N \leq 400$. Za svaki par različitih otoka x i y ($0 \leq x < y \leq N - 1$), postoje točno dva kanua koji mogu biti korišteni za putovanje između tih otoka. Jedan od njih usidren je na otoku x , a drugi je usidren na otoku y .

3. (21 bod) $N \leq 1000$, M je paran, i za svaki **paran** i takav da je $0 \leq i \leq M - 1$, kanui i i $i + 1$ mogu se koristiti za putovanje između otoka $U[i]$ i $V[i]$. Kanu i inicijalno je usidren na otoku $U[i]$, a kanu $i + 1$ inicijalno je usidren na otoku $V[i]$. Formalno, $U[i] = V[i + 1]$ i $V[i] = U[i + 1]$.
4. (24 boda) $N \leq 1000$, M je paran, i za svaki **paran** i takav da je $0 \leq i \leq M - 1$, kanui i i $i + 1$ mogu se koristiti za putovanje između otoka $U[i]$ i $V[i]$. Oba kanua inicijalno su usidrena na otoku $U[i]$. Formalno, $U[i] = U[i + 1]$ i $V[i] = V[i + 1]$
5. (45 bodova) Nema dodatnih ograničenja.

Za svaki test podatak u kojem postoji romantično putovanje, vaše rješenje:

- osvaja sve bodove ako pronađe romantično putovanje,
- osvaja 35% bodova ako vrati true, polje dulje od 2 000 000 elemenata, ili polje koje ne odgovara romantičnom putovanju,
- u protivnom osvaja 0 bodova.

Za svaki test podatak u kojem ne postoji romantično putovanje, vaše rješenje:

- osvaja sve bodove ako vrati false,
- u protivnom osvaja 0 bodova.

Primijetite da je konačan broj bodova nekog podzadatka zapravo najmanji broj bodova osvojenih na nekom test podatku u tom podzadatku.

Ogledni ocjenjivač

Ogledni ocjenjivač čita ulazne podatke u sljedećem formatu:

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

Ogledni ocjenjivač ispisuje vaše odgovore u sljedećem formatu:

- Ako `find_journey` vrati `bool`:
 - redak 1: 0
 - redak 2: 0 ako `find_journey` vrati `false`, ili 1 inače.
- Ako `find_journey` vrati `int []`, označimo elemente tog polja s $c[0], c[1], \dots, c[k - 1]$. Ogledni ocjenjivač ispisuje:
 - redak 1: 1
 - redak 2: k
 - redak 3: $c[0] c[1] \dots c[k - 1]$