



## Zájazd na Seriby

Seriby sú skupinka ostrovov nachádzajúcich sa v Jávskom mori. Skladajú sa z  $N$  ostrovov, očíslovaných od 0 po  $N - 1$ .

Na ostrovoch sa nachádza  $M$  lodičiek, očíslovaných od 0 po  $M - 1$ . Dá sa nimi prepravovať medzi ostrovmi. Lodička číslo  $i$  ( $0 \leq i \leq M - 1$ ) môže byť zaparkovaná buď na ostrove  $U[i]$  alebo na ostrove  $V[i]$ . Každá lodička môže byť použitá len na presun medzi svojimi dvoma ostrovmi. Prirodzene, keď z ostrova  $U[i]$  lodičku odparkujeme a preplávame ňou na ostrov  $V[i]$ , lodička ostane zaparkovaná na ostrove  $V[i]$ , a naopak.

Lodička  $i$  je na začiatku zaparkovaná na ostrove  $U[i]$ . Môže sa stať, že medzi dvoma ostrovmi premáva viac ako jedna lodička. Na každom ostrove môže byť v každej chvíli zaparkovaných ľubovoľne veľa lodičiek.

Z bezpečnostných dôvodov musí byť každá lodička servisovaná po každej jednej jazde. Preto platí, že žiadnu lodičku nesmieme použiť dvakrát po sebe - medzi každými dvoma jazdami tou istou lodičkou musí byť aspoň jedna jazda inou lodičkou.

Teta Dengklek si plánuje lodičkový zájazd naprieč Seribskými ostrovmi. Jej trasa je **schodná** vtedy a iba vtedy ak sú splnené nasledujúce podmienky:

- Zájazd začína a končí na ostrove 0.
- Počas zájazdu je navštívený aspoň jeden ostrov iný ako ostrov 0.
- Po konci zájazdu sú všetky lodičky vrátené a zaparkované na svojich pôvodných miestach. To znamená, že každá lodička  $i$  ( $0 \leq i \leq M - 1$ ) musí byť na konci zájazdu zaparkovaná na ostrove  $U[i]$

Pomôžte Tete Dengklek vymyslieť **schodný** zájazd, v ktorom bude najviac 2 000 000 plavieb na lodičkách, alebo zistite, že žiaden taký zájazd neexistuje. Pri obmedzeniach v tejto úlohe sa dá dokázať, že ak existuje nejaký **schodný** zájazd, tak existuje **schodný** zájazd, ktorý nevyžaduje viac ako 2 000 000 plavieb.

## Implementačné Detaily

Implementujte nasledovnú funkciu:

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

- $N$ : počet ostrovov.
- $M$ : počet lodičiek.
- $U, V$ : polia dĺžky  $M$  popisujúce lodičky.
- Táto funkcia by mala navrátiť buď boolovskú hodnotu alebo pole celých čísiel
  - Ak neexistuje žiaden schodný zájazd, funkcia musí vrátiť `false`.
  - Ak schodný zájazd existuje, sú dve možnosti:
    - Aby ste obdržali plné skóre, funkcia musí vrátiť pole najviac 2 000 000 celých čísiel reprezentujúce schodný zájazd. (Prvky tohto poľa majú byť čísla lodičiek v poradí, v ktorom sa nimi budete plaviť.)
    - Aby ste obdržali čiastočné body, funkcia musí vrátiť `true`, alebo pole obsahujúce viac ako 2 000 000 čísiel, alebo pole čísiel nepopisujúcich schodný zájazd. (Viac detailov o bodovaní je v časti Podúlohy.)
- Táto funkcia je volaná práve raz.

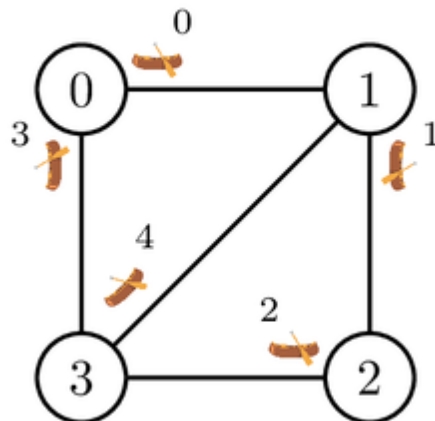
## Príklady

### Príklad 1

Pozrime sa na nasledovné volanie:

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

Ostrov a lodičky sú zobrazené na obrázku:



Jeden schodný zájazd vyzerá nasledovne. Teta Dengklek sa na začiatok bude postupne plaviť lodičkami 0, 1, 2 a 4. Touto postupnosťou skončí na ostrove 1. Potom môže Teta Dengklek použiť lodičku 0 znovu, nakoľko je zaparkovaná na ostrove 1 a posledná lodička, ktorú použila, nie je lodička 0. Po tejto plavbe na lodičke 0 bude Teta Dengklek na ostrove 0. Nanešťastie, lodičky 1, 2 a 4 nie sú zaparkované na svojich pôvodných ostrovoch. Zájazd teda pokračuje, a to lodičkami 3, 2, 1, 4

a opäť 3. Teta Dengklek sa potom znovu nachádza na ostrove 0, tentokrát už ale zanechávajú všetky lodičky zaparkované na správnych miestach.

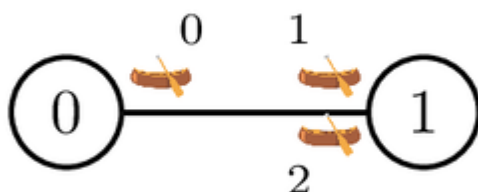
Popis tohoto zájazdu - pole  $[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]$  - je teda jedným zo správnych výstupov, ktoré môže naša funkcia vrátiť.

## Príklad 2

Prejdime si ešte druhé volanie:

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

Ostrov a lodičky sú zobrazené na obrázku:



Teta Dengklek nemá inú možnosť, iba začať plavbu lodičkou 0. Po nej má na výber z lodičiek 1 a 2. (Nezabúdajte, že nemôže hneď znovu použiť lodičku 0.) Bez ohľadu na to, či si vyberie lodičku 1 alebo 2, skončí po plavbe ňou znovu na ostrove 0. Je teda späť na začiatku, ale na jej veľké nešťastie platí, že niektoré lodičky nie sú zaparkované na správnych ostrovoch. A čo je ešte horšie, teraz už ani nemáme žiadnu lodičku, ktorú by sme mohli použiť, lebo jediná lodička na ostrove, na ktorom sa nachádzame, je tá, ktorou sme práve prišli.

Neexistuje teda žiaden schodný zájazd, čiže naša funkcia musí vrátiť `false`.

## Obmedzenia

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 200\,000$
- $0 \leq U[i] \leq N - 1$  a  $0 \leq V[i] \leq N - 1$  (pre každé  $i$  také, že  $0 \leq i \leq M - 1$ )
- $U[i] \neq V[i]$  (pre každé  $i$  také, že  $0 \leq i \leq M - 1$ )

## Podúlohy

1. (5 bodov)  $N = 2$

2. (5 bodov)  $N \leq 400$ . Pre každý pár rôznych ostrovov  $x$  a  $y$  ( $0 \leq x < y \leq N - 1$ ) existujú práve dve lodičky, ktoré môžu byť použité na plavbu medzi nimi. Jedna z nich je zaparkovaná na ostrove  $x$ , druhá na ostrove  $y$ .
3. (21 bodov)  $N \leq 1000$ ,  $M$  je párne a pre každé **párne**  $i$  také, že  $0 \leq i \leq M - 1$ , lodičky  $i$  a  $i + 1$  sa môžu obe plaviť medzi ostrovmi  $U[i]$  a  $V[i]$ . Lodička  $i$  je počiatočne zaparkovaná na ostrove  $U[i]$  a lodička  $i + 1$  na ostrove  $V[i]$ . Formálnejšie,  $U[i] = V[i + 1]$  a  $V[i] = U[i + 1]$ .
4. (24 bodov)  $N \leq 1000$ ,  $M$  je párne a pre každé párne  $i$  také, že  $0 \leq i \leq M - 1$ , lodičky  $i$  a  $i + 1$  sa môžu obe plaviť medzi ostrovmi  $U[i]$  a  $V[i]$ . Obe lodičky sú počiatočne zaparkované na ostrove  $U[i]$ . Formálnejšie,  $U[i] = U[i + 1]$  a  $V[i] = V[i + 1]$ .
5. (45 bodov) Žiadne ďalšie obmedzenia.

Pre každý testovací vstup, v ktorom existuje schodný zjazd, vaše riešenie:

- dostane plný počet bodov ak vráti schodný zjazd,
- dostane 35% bodov ak vráti true, alebo pole s viac než 2 000 000 číslami, alebo pole nepopisujúce schodný zjazd,
- 0 bodov vo všetkých ostatných prípadoch.

Pre každý testovací vstup, v ktorom schodný zjazd neexistuje, vaše riešenie:

- dostane plný počet bodov, ak vráti false,
- 0 bodov vo všetkých ostatných prípadoch.

Výsledné skóre pre každú podúlohu je minimum bodov zo všetkých testovacích vstupov v danej podúlohe

## Lokálny Testovač

Lokálny testovač číta vstup v nasledovnom formáte:

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

Lokálny testovač vypisuje odpovede v nasledovnom formáte:

- Ak `find_journey` vrátila `bool`:
  - riadok 1: 0
  - riadok 2: 0 ak `find_journey` vrátila `false`, 1 vo všetkých ostatných prípadoch.
- Ak `find_journey` vrátila `int[]`, označme si prvky tohoto pola  $c[0], c[1], \dots, c[k - 1]$ . Lokálny testovač vypíše:
  - riadok 1: 1
  - riadok 2:  $k$
  - riadok 3:  $c[0] c[1] \dots c[k - 1]$