



## Podjela atrakcija

Baku ima  $n$  turističkih atrakcija, numerisanih od 0 do  $n - 1$ . Ove atrakcije su povezane sa ukupno  $m$  dvosmjernih ulica, numerisanih od 0 do  $m - 1$ . Svaka ulica povezuje dvije različite atrakcije i **moгуće je doći iz svake od njih do bilo koje druge koristeći ove ulice**.

Fatima planira da posjeti sve ove atrakcije tokom tri dana. Odlučila je da podijeli svih  $n$  atrakcija u tri skupa  $A$ ,  $B$  i  $C$  veličina  $a$ ,  $b$  i  $c$ . Svaka od atrakcija će pripadati tačno jednom od tri skupa,  $a + b + c = n$ .

Fatima bi željela da nađe skupove  $A$ ,  $B$  i  $C$  tako da su **najmanje dva** od njih tri **povezana**. Za skup  $S$  atrakcija kažemo da je povezan ako je moguće putovati između bilo koje dvije atrakcije u skupu  $S$  koristeći date putevi a da pri tome nikada ne morate doći u atrakciju koja nije u skupu  $S$  (odnosno izaći iz skupa  $S$ ). Jedna podjela atrakcija na skupove  $A$ ,  $B$  i  $C$  je **validna** ako zadovoljava gore navedene uslove.

Pomozite Fatimi da pronađe validnu podjelu atrakcija (za date  $a$ ,  $b$  i  $c$ ), ili se uvjerite da takva podjela ne postoji. Ukoliko postoji više validnih podjela odredite bilo koju od njih.

## Detalji implementacije

Potrebno je implementirati sljedeću proceduru:

```
int[] find_split(int n, int a, int b, int c, int[] p, int[] q)koliko
```

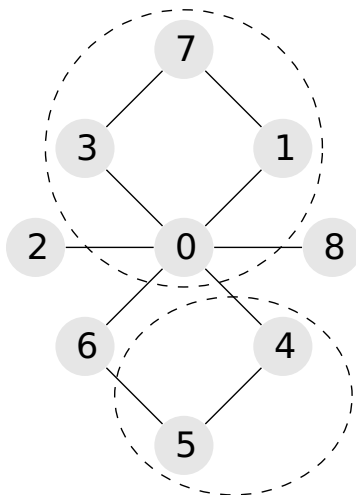
- $n$ : broj atrakcija u Bakuu.
- $a$ ,  $b$ , i  $c$ : broj atrakcija koje treba posjetiti redom prvog, drugog i trećeg dana.
- $p$  i  $q$ : nizovi dužine  $m$ , koji predstavljaju indekse atrakcija na krajevima odgovarajućih ulica. Za svako  $i$  ( $0 \leq i \leq m - 1$ ),  $p[i]$  i  $q[i]$  su atrakcije povezane ulicom  $i$ .
- Procedura treba da vrati niz dužine  $n$ , označimo ga sa  $s$ . Ako ne postoji podjela tako da je posjeta uspješna, niz  $s$  treba da sadrži  $n$  nula. Inače, za  $0 \leq i \leq n - 1$ ,  $s[i]$  treba da ima vrijednost 1, 2, ili 3 i da označava, respektivno, kojem od skupova  $A$ ,  $B$  i  $C$  atrakcija  $i$  je dodjeljena.

## Primjeri

## Primjer 1

Posmatrajmo sljedeći poziv:

```
find_split(9, 4, 2, 3, [0, 0, 0, 0, 0, 0, 1, 3, 4, 5],  
           [1, 2, 3, 4, 6, 8, 7, 7, 5, 6])
```

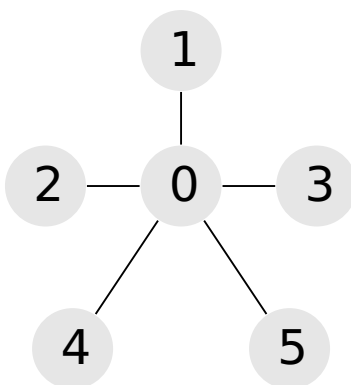


Jedno od mogućih validnih rješenja je  $[1, 1, 3, 1, 2, 2, 3, 1, 3]$ . Ovo rješenje opisuje sljedeću podjelu:  $A = \{0, 1, 3, 7\}$ ,  $B = \{4, 5\}$ , i  $C = \{2, 6, 8\}$ . Skupovi  $A$  i  $B$  su povezani (pogledati sliku).

## Primjer 2

Posmatrajmo sljedeći poziv:

```
find_split(6, 2, 2, 2, [0, 0, 0, 0, 0], [1, 2, 3, 4, 5])
```



Ne postoji podjela za uspešnu posjetu atrakcija, tako da je jedino validno rješenje  $[0, 0, 0, 0, 0, 0]$ .

## Ograničenja

- $3 \leq n \leq 100\,000$
- $2 \leq m \leq 200\,000$
- $1 \leq a, b, c \leq n$
- $a + b + c = n$
- Postoji najviše jedna ulica između svake dvije atrakcije.
- Moguće je doći iz svake atrakcije do bilo koje druge koristeći postojeće ulice.
- $0 \leq p[i], q[i] \leq n - 1$  i  $p[i] \neq q[i]$  za  $0 \leq i \leq m - 1$

## Podzadaci

1. (7 points) Svaka atrakcija se nalazi na krajevima najviše dvije ulice
2. (11 points)  $a = 1$
3. (22 points)  $m = n - 1$
4. (24 points)  $n \leq 2500, m \leq 5000$
5. (36 points) Nema dodatnih ograničenja.

## Grader

Grader učitava podatke u sljedećem formatu:

- linija 1:  $n \ m$
- linija 2:  $a \ b \ c$
- linija  $3 + i$  (za  $0 \leq i \leq m - 1$ ):  $p[i] \ q[i]$

Grader ispisuje jednu liniju sa nizom koji vraća procedura `find_split`.