



Podjela naselja

Nakon osvojene medalje na takmičenju, Matija se pobjednički vratio u svoj grad Podgoricu. Podgorica ima n naselja, numerisanih brojevima od 0 do $n - 1$. Naselja su povezana sa ukupno m dvosmjernih ulica, numerisanih brojevima od 0 do $m - 1$. Svaka ulica povezuje dva različita naselja i **moguće je doći iz svakog naselja do bilo kog drugog koristeći te ulice**.

Pošto nije bio moguć doček na balkonu zbog njegovog malog kapaciteta, Matija je odlučio da posjeti **svako naselje tačno jednom** u naredna tri dana i podijeli radost sa građanima Podgorice.

Prvog dana Matija će posjetiti skup od a naselja, drugog dana skup od b i trećeg dana skup od c naselja. Primjetimo da je zbir veličina ova tri skupa tačno n ($a + b + c = n$).

Da bi celokupna posjeta bila uspješna, potrebno je da **bar dva** skupa budu povezana. Za jedan skup naselja kažemo da je povezan, ako postoji put između svaka dva naselja u tom skupu koji prolazi samo kroz naselje iz tog skupa.

Odrediti jednu od mogućih podjela naselja u skupove, tako da Matijina posjeta bude uspješna, ako takva podjela postoji.

Detalji implementacije

Potrebno je implementirati sljedeću funkciju:

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

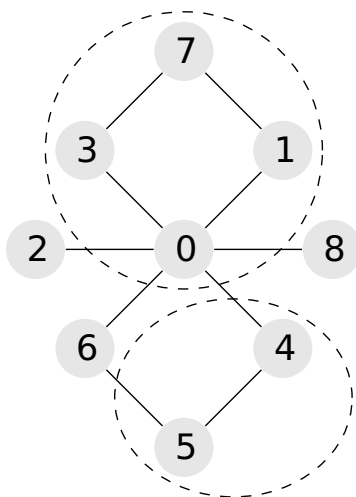
- n : broj naselja u Podgorici.
- a , b , i c : broj naselja koji treba posjetiti redom prvog, drugog i trećeg dana.
- p i q : nizovi dužine m , koji predstavljaju indekse naselja na krajevima odgovarajućih ulica. Za svako i ($0 \leq i \leq m - 1$), $p[i]$ i $q[i]$ naselja su povezana ulicom i .
- Funkcija treba da vrati niz dužine n , označimo ga sa s . Ako ne postoji podjela tako da posjeta bude 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 označava dan kada Matija treba da posjeti naselje sa indeksom i .

Primjeri

Primjer 1

Posmatrajmo sljedeći poziv funkcije:

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

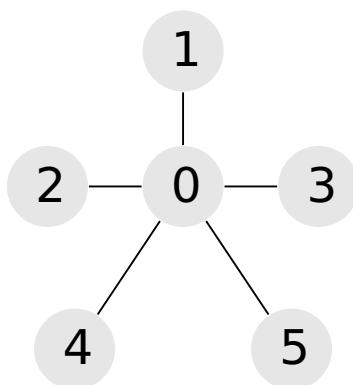


Jedno od mogućih validnih rješenja je $[1, 1, 3, 1, 2, 2, 3, 1, 3]$. Prvog dana Matija će posjetiti skup naselja: $\{0, 1, 3, 7\}$, drugog dana skup naselja $\{4, 5\}$ i trećeg dana skup naselja $\{2, 6, 8\}$. Skupovi naselja koji su posjećeni prvog dana i drugog dana su povezani (pogledati sliku).

Primjer 2

Posmatrajmo sledeći poziv funkcije:

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



Ne postoji podjela za uspješnu posjetu naselja, 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 svaka dva naselja
- Moguće je doći iz svakog naselja do bilo kog drugog koristeći postojeće ulice
- $0 \leq p[i], q[i] \leq n - 1$ i $p[i] \neq q[i]$ for $0 \leq i \leq m - 1$

Podzadaci

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

Program za ocjenjivanje (grader)

Program za ocjenjivanje (grader) učitava podatke u sljedećem formatu:

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

Program za ocjenjivanje (grader) štampa jedan red koji sadrži niz koji vraća funkcija `find_split`.