Split the Attractions

Există n atracții în Baku, numerotate de la 0 la $n-1$. De asemenea, există m drumuri bidirecționale, numerotate de la 0 la $m-1$. Fiecare drum leagă două atracții diferite. Se poate călători între oricare două atracții folosind drumurile date.

Fatima planifică să viziteze toate atracțiile în trei zile. Ea va împărți cele n atracții în trei mulțimi A, B și C de dimensiuni a, b respectiv c. Fiecare atracție va aparține exact unei mulțimi, deci $a+b+c=n$.

Fatima dorește să identifice mulțimile A, B și C, astfel încât **cel puțin două** din cele trei mulțimi să fie **conectate**. O mulțime S de atracții se numește conectată dacă în mulțimea S există posibilitatea de a călători între oricare două atracții utilizând drumuri, fără a trece prin atracții ce nu fac parte din S. O împărțire a atracțiilor în mulțimile A, B și C se numește **validă** dacă ea satisface condițiile descrise mai sus.

Ajutați-o pe Fatima să găsească o împărțire validă a atracțiilor (cunoscându-se a, b și c), sau să determine dacă o astfel de împărțire nu există. Dacă există mai multe împărțiri valide, puteți găsi oricare din ele.

Detalii de implementare

Trebuie să implementați următoarea funcție:

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

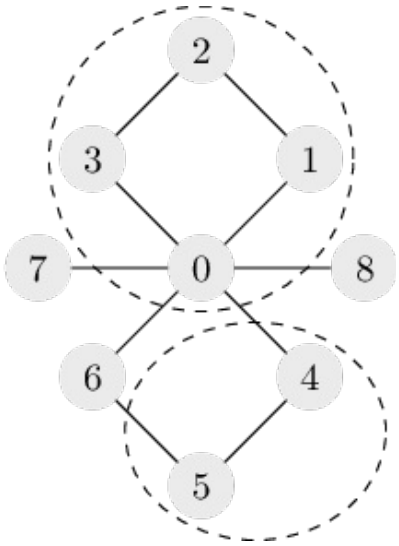
- n: numărul de atracții.
- a, b, și c: dimensiunile dorite ale mulțimilor A, B și C.
- p și q: vectori de lungime m, care conțin extremitățile drumurilor. Pentru fiecare i ($0 \leq i \leq m-1$), $p[i]$ și $q[i]$ sunt două atracții conectate prin drumul i.
- Această procedură va returna un vector de lungime n. Notăm acest vector prin s. Dacă nu există o împărțire validă, atunci s va conține n zerouri. Altfel, pentru fiecare $0 \leq i \leq n-1$, $s[i]$ trebuie să fie una din valorile 1, 2, sau 3 însemnând că atracția i este atribuită mulțimii A, B respectiv C.

Exemple

Exemplul 1

Se consideră următorul apel:

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

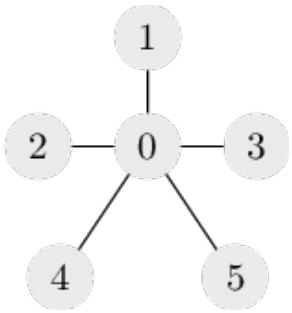


O posibilă soluție corectă este $[1, 1, 1, 1, 2, 2, 3, 3, 3]$. Această soluție descrie următoarea împărțire: $A = 0, 1, 2, 3$, $B = 4, 5$, și $C = 6, 7, 8$. Mulțimile A și B sunt conectate.

Exemplul 2

Se consideră următorul apel:

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



Nu există împărțiri valide. Prin urmare, răspunsul corect este $[0, 0, 0, 0, 0, 0]$.

Restricții

- $3 \leq n \leq 100\,000$
- $2 \leq m \leq 200\,000$
- $1 \leq a, b, c \leq n$
- $a + b + c = n$
- Există cel mult un drum între oricare două atracții.

- Există posibilitatea de a călători între oricare două atracții utilizând drumurile date.
- $0 \leq p[i], q[i] \leq n - 1$ și $p[i] \neq q[i]$ pentru $0 \leq i \leq m - 1$

Subtask-uri

1. (7 puncte) Fiecare atracție reprezintă extremitatea a cel puțin două drumuri.
2. (11 puncte) $a = 1$
3. (22 de puncte) $m = n - 1$
4. (24 de puncte) $n \leq 2500, m \leq 5000$
5. (36 de puncte) Fără restricții suplimentare.

Exemplu de grader

Grader-ul citește din input în următorul format:

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

Grader-ul afișează pe o singură linie vectorul returnat de `find_split`.