



Raggruppa le attrazioni

Ci sono n attrazioni in Baku, numerate da 0 a $n - 1$. Ci sono anche m strade che possono essere percorse in entrambe le direzioni, numerate da 0 a $m - 1$. Ogni strada collega due attrazioni differenti. Esiste almeno un percorso valido tra ogni coppia di attrazioni.

Fatima vuole visitare tutte le attrazioni in 3 giorni. Per farlo, deve raggruppare le n attrazioni in tre insiemi differenti A , B , e C di dimensioni rispettivamente a , b , e c . Ogni attrazione farà parte di esattamente un insieme, per cui $a + b + c = n$.

Fatima vuole scegliere gli insiemi A , B , e C , di modo che **almeno due** dei tre insiemi siano **connessi**. Un insieme di attrazioni S è connesso se è possibile viaggiare tra tutte le attrazioni in S utilizzando le strade presenti e senza mai passare per un'attrazione che non faccia parte di S . Una partizione delle attrazioni negli insiemi A , B , e C è considerata **valida** se soddisfa tutte le condizioni descritte in precedenza.

Aiuta Fatima a trovare una *qualunque* partizione valida delle attrazioni (conoscendo a , b , e c) o ad appurare che non ne esiste alcuna.

Dettagli di implementazione

Devi implementare la seguente funzione:

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

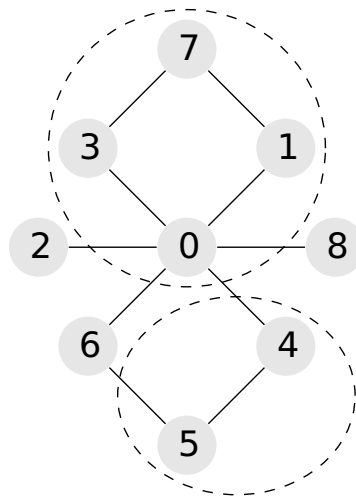
- n : il numero di attrazioni.
- a , b , e c : le dimensioni degli insiemi A , B , e C .
- p e q : array di lunghezza m , che contengono le attrazioni agli estremi della strada. Per ogni i ($0 \leq i \leq m - 1$), $p[i]$ e $q[i]$ sono le due attrazioni connesse dalla strada i .
- La funzione deve restituire un array di lunghezza n , chiamato s d'ora in poi. Se non esiste una partizione valida, s deve contenere n zero. Altrimenti $s[i]$ dovrebbe contenere 1, 2, oppure 3 (per $0 \leq i \leq n - 1$) per indicare che l'attrazione i è assegnata rispettivamente all'insieme A , B , o C .

Esempi

Esempio 1

Considera la seguente chiamata:

```
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])
```

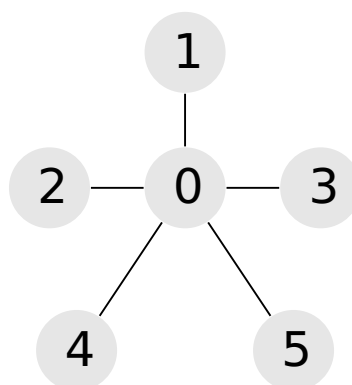


Una soluzione possibile è $[1, 1, 3, 1, 2, 2, 3, 1, 3]$. Questa soluzione è associata alla seguente partizione: $A = \{0, 1, 3, 7\}$, $B = \{4, 5\}$, e $C = \{2, 6, 8\}$ in cui gli insiemi A e B sono connessi.

Esempio 2

Considera la seguente chiamata:

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



In questo caso non esiste nessuna partizione valida: quindi, l'unica risposta corretta è $[0, 0, 0, 0, 0, 0]$.

Assunzioni

- $3 \leq n \leq 100\,000$.
- $2 \leq m \leq 200\,000$.
- $1 \leq a, b, c \leq n$.
- $a + b + c = n$.
- Ogni coppia di attrazioni è collegata al massimo da una strada.
- Ogni attrazione è raggiungibile partendo da qualsiasi altra.
- $0 \leq p[i], q[i] \leq n - 1$ e $p[i] \neq q[i]$ for $0 \leq i \leq m - 1$.

Subtask

1. (7 punti) Ogni attrazione è collegata al massimo a due strade.
2. (11 punti) $a = 1$.
3. (22 punti) $m = n - 1$.
4. (24 punti) $n \leq 2500, m \leq 5000$.
5. (36 punti) Nessuna limitazione aggiuntiva.

Grader di esempio

Il grader di esempio legge l'input nel seguente formato:

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

Il grader di esempio stampa un'unica riga contenente l'array restituito da `find_split`.