



Aufteilung der Attraktionen

Es gibt n Attraktionen in Baku die von 0 bis $n - 1$ nummeriert sind. Darüber hinaus gibt es m in beide Richtungen befahrbare Strassen die von 0 bis $m - 1$ nummeriert sind. Jede Strasse verbindet zwei verschiedene Attraktionen. Man kann von jeder Attraktion aus jede andere Attraktion erreichen, indem man das Strassennetz benutzt.

Fatima plant alle Attraktionen in drei Tagen zu besichtigen. Sie hat sich entschieden am ersten Tag a Attraktionen, am zweiten Tag b Attraktionen und am dritten Tag c Attraktionen zu besuchen. Deshalb will sie die Attraktionen in drei Mengen A , B und C der Grösse a , b und c aufteilen. Jede Attraktion soll in genau einer Menge sein, folglich gilt $a + b + c = n$.

Fatima würde gerne die Mengen A , B und C so bestimmen, dass **mindestens zwei** der drei Mengen **zusammenhängend** sind. Eine Menge von Attraktionen S ist **zusammenhängend**, wenn es möglich ist, über das Strassennetz von jeder Attraktion in S zu jeder anderen Attraktion in S zu gelangen, ohne bei einer Attraktion, die nicht in S ist, vorbeizukommen. Eine Aufteilung der Attraktionen in Mengen A , B und C ist **gültig**, falls sie die oben beschriebenen Bedingungen erfüllt.

Hilf Fatima eine gültige Aufteilung der Attraktionen zu finden (a , b und c sind gegeben) oder stelle fest, dass es keine gibt. Falls es mehrere gültige Aufteilungen gibt, kannst du von diesen eine beliebige zurückgeben.

Implementierung

Implementiere die folgende Funktion:

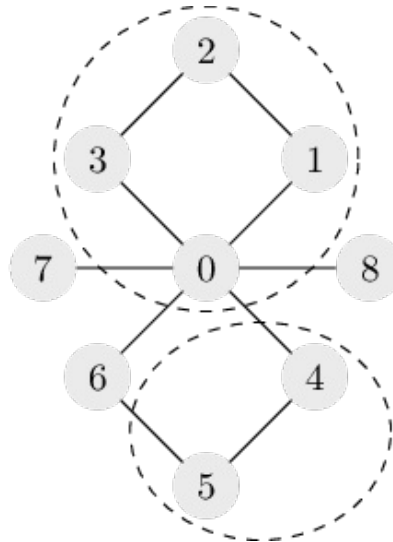
```
int[] find_split(int n, int a, int b, int c, int[] p, int[] q)
```

- n : die Anzahl der Attraktionen.
- a , b und c : die Grössen der Mengen A , B und C .
- p und q : Arrays der Länge m , die die Endpunkte der Strassen enthalten. Für jedes i ($0 \leq i \leq m - 1$) sind $p[i]$ und $q[i]$ die beiden Attraktionen die durch die Strasse i verbunden sind.
- Die Funktion soll ein Array der Länge n zurückgeben. Wir nennen dieses Array s . Wenn es keine gültige Aufteilung gibt, soll s aus n Nullen bestehen. Andernfalls, für $0 \leq i \leq n - 1$, soll $s[i]$ entweder 1, 2 oder 3 sein, um anzugeben, zu welcher Menge die Attraktion i gehört: 1 bedeutet A , 2 bedeutet B und 3 bedeutet C .

Beispiele

Beispiel 1

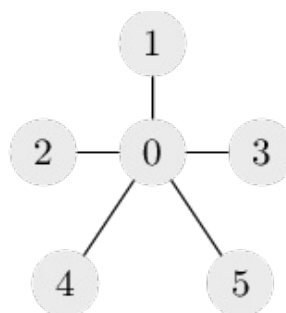
```
find_split(9, 4, 2, 3, [0, 0, 0, 0, 0, 0, 1, 2, 4, 5],  
           [1, 3, 4, 6, 7, 8, 2, 3, 5, 6])
```



Eine mögliche Lösung ist $[1, 1, 1, 1, 2, 2, 3, 3, 3]$. Diese Lösung beschreibt die folgende Aufteilung: $A = \{0, 1, 2, 3\}$, $B = \{4, 5\}$ und $C = \{6, 7, 8\}$. Die Mengen A und B sind zusammenhängend.

Beispiel 2

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



Es gibt keine gültige Aufteilung. Die einzige korrekte Antwort ist folglich $[0, 0, 0, 0, 0, 0]$.

Beschränkungen

- $3 \leq n \leq 100\,000$
- $2 \leq m \leq 200\,000$
- $1 \leq a, b, c \leq n$

- $a + b + c = n$
- Es gibt höchstens eine Strasse zwischen zwei Attraktionen.
- Es ist möglich von jeder Attraktion aus jede andere über das Strassennetz zu erreichen.
- $0 \leq p[i], q[i] \leq n - 1$ und $p[i] \neq q[i]$ für $0 \leq i \leq m - 1$

Subtasks

1. (7 Punkte) Jede Attraktion ist Endpunkt von höchstens zwei Strassen.
2. (11 Punkte) $a = 1$
3. (22 Punkte) $m = n - 1$
4. (24 Punkte) $n \leq 2500, m \leq 5000$
5. (36 Punkte) Keine zusätzlichen Beschränkungen.

Beispiel-Grader

Der Beispiel-Grader liest die Eingabe im folgenden Format:

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

Der Beispiel-Grader gibt eine einzelne Zeile mit dem von `find_split` zurückgegebenen Array aus.