



Divide las atracciones

Hay n atracciones en Baku, numeradas de 0 a $n - 1$. Hay además m caminos de doble vía, numerados de 0 a $m - 1$. Cada camino conecta dos atracciones diferentes. Es posible viajar entre cualquier par de atracciones a través de los caminos.

Fátima está planeando visitar todas las atracciones en tres días. Ella ya decidió que quiere visitar a atracciones en el primer día, b atracciones en el segundo día, y c atracciones en el tercer día. Por lo tanto, ella dividirá las n atracciones en tres conjuntos A , B , y C de tamaños a , b , y c , respectivamente. Cada atracción pertenecerá a exactamente un conjunto, entonces $a + b + c = n$.

A Fátima le gustaría encontrar los conjuntos A , B y C , tal que **al menos dos** de los tres conjuntos estén **conectados**. Un conjunto S de atracciones es llamado conectado si es posible viajar entre cualquier par de atracciones en S usando los caminos y sin pasar por ninguna atracción que no esté en S . Una división de atracciones en los conjuntos A , B y C es llamada **válida** si esta satisface las condiciones descritas a continuación.

Ayuda a Fátima a encontrar una división válida de las atracciones (dados a , b , y c), o determina que no existe una división válida. Si hay múltiples divisiones válidas, puedes encontrar cualquiera de ellas.

Detalles de implementación

Debes implementar el siguiente procedimiento:

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

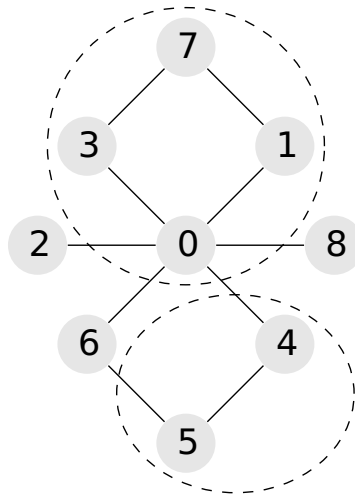
- n : el número de atracciones.
- a , b , y c : los tamaños deseados de los conjuntos A , B , y C .
- p y q : arreglos de tamaño m , conteniendo los puntos finales de los caminos. Para cada i ($0 \leq i \leq m - 1$), $p[i]$ y $q[i]$ son las dos atracciones conectadas por el camino i .
- Este procedimiento debe retornar un arreglo de tamaño n . Denote el arreglo por s . Si no hay una división válida, s debe contener n ceros. De otra manera, para $0 \leq i \leq n - 1$, $s[i]$ debe ser uno de 1, 2, or 3 para denotar que la atracción i está asignada al conjunto A , B , y C , respectivamente.

Ejemplos

Ejemplo 1

Considere la siguiente llamada:

```
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])
```

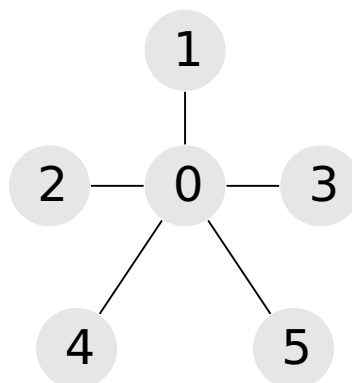


Una posible solución correcta es $[1, 1, 3, 1, 2, 2, 3, 1, 3]$. Esta solución describe la siguiente división: $A = \{0, 1, 3, 7\}$, $B = \{4, 5\}$, y $C = \{2, 6, 8\}$. Los conjuntos A y B están conectados.

Ejemplo 2

Considere la siguiente llamada:

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



No existe una división válida. Por lo tanto, la única respuesta correcta es $[0, 0, 0, 0, 0, 0]$.

Restricciones

- $3 \leq n \leq 100\,000$
- $2 \leq m \leq 200\,000$
- $1 \leq a, b, c \leq n$
- $a + b + c = n$
- Hay a lo mucho un camino entre cada par de atracciones.
- Es posible viajar entre cualquier par de atracciones a través de los caminos.
- $0 \leq p[i], q[i] \leq n - 1$ y $p[i] \neq q[i]$ para $0 \leq i \leq m - 1$

Subtareas

1. (7 puntos) Cada atracción es un punto final de a lo mucho dos caminos.
2. (11 puntos) $a = 1$
3. (22 puntos) $m = n - 1$
4. (24 puntos) $n \leq 2500, m \leq 5000$
5. (36 puntos) Sin restricciones adicionales.

Grader de muestra

El grader de muestra lee la entrada en el siguiente formato:

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

El grader de muestra imprime una línea conteniendo el arreglo retornado por `find_split`.