



Divisiones de atracciones

En Baku hay n atracciones numeradas de 0 a $n - 1$. También hay m caminos bidireccionales numerados de 0 a $m - 1$. Cada camino conecta dos atracciones diferentes. Es posible viajar entre cualquier par de atracciones usando los caminos.

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

Fátima quiere encontrar los conjuntos A , B y C de tal forma que **al menos** dos de los tres conjuntos estén **conectados**. Un conjunto S de atracciones está conectado si es posible viajar entre cualesquiera par de atracciones en S usando caminos sin pasar por alguna atracción que no se encuentre en S . Una división de atracciones en conjuntos A , B y C es considerada **válida** si satisface las condiciones mencionadas anteriormente.

Ayuda a Fátima a encontrar una división válida de las atracciones (dado a , b y c) o determinar si la división no existe. Si hay varias 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)
```

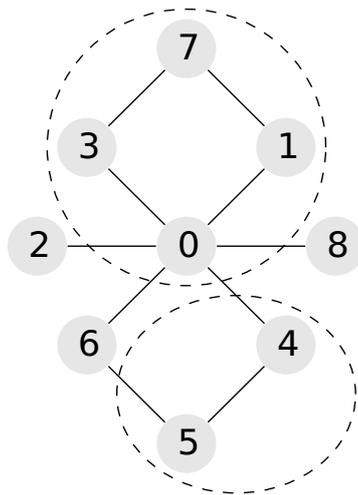
- n : el número de atracciones.
- a , b , y c : el tamaño deseado de los conjuntos A , B , y C .
- p y q : arreglos de longitud m , contiene las atracciones que conectan los caminos. Para cada i ($0 \leq i \leq m - 1$), $p[i]$ y $q[i]$ son las dos atracciones conectadas por el camino i .
- El procedimiento debe regresar un arreglo de tamaño n . El arreglo se denota como s . Si no existe una división válida, s debe contener n ceros. De lo contrario, para $0 \leq i \leq n - 1$, $s[i]$ debe ser 1, 2, o 3 para denotar que la atracción i es asignada al conjunto A , B , o C , respectivamente.

Ejemplos

Ejemplo 1

Considera la siguiente llamada:

```
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])
```

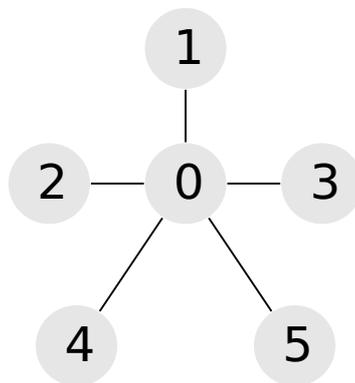


Una posible solución es $[1, 1, 1, 1, 2, 2, 3, 3, 3]$. Esta solución describe la división: $A = 0, 1, 2, 3$, $B = 4, 5$, y $C = 6, 7, 8$. Los conjuntos A y B son conectados.

Ejemplo 2

Considera la siguiente llamada:

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



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

Consideraciones

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

Subtarea

1. (7 puntos) Cada atracción es el punto final de a lo más 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.

Evaluador de ejemplo

El evaluador de ejemplo 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 evaluador de ejemplo imprime una línea con el arreglo regresado por `find_split`.