Llaves

El arquitecto Timothy ha diseñado un juego del tipo cuarto de escape. En este juego, hay n cuartos numerados de 0 a $n-1$. Inicialmente, cada cuarto contiene exactamente una llave. Cada llave es de un tipo, el cual es un entero entre 0 y $n-1$, inclusive. El tipo de la llave en el cuarto i ($0 \le i \le n-1$) es $r[i]$. Nótese que múltiples cuartos pueden contener llaves del mismo tipo, es decir, los valores $r[i]$ no son necesariamente distintos.

También existen m conectores **bidireccionales** en el juego, numerados de 0 a $m-1$. El conector j ($0 \le j \le m-1$) conecta un par de cuartos distintos $u[j]$ y $v[j]$. Un par de cuartos pueden ser conectados por múltiples conectores.

El juego es jugado por un solo jugador que recoge las llaves y se mueve entre los cuartos recorriendo los conectores. Decimos que el jugador **recorre** el conector j cuando usa este conector para moverse del cuarto $u[j]$ al cuarto $v[j]$, o viceversa. El jugador solo puede recorrer el conector j si ha recogido una llave de tipo $c[j]$ previamente.

En todo momento del juego, el jugador se encuentra en algún cuarto x y puede realizar dos tipos de acciones:

- recoger la llave del cuarto x, cuyo tipo es $r[x]$ (a menos que ya la haya recogido antes),
- recorrer el conector j, donde $u[j] = x$ o $v[j] = x$, si el jugador ha recogido una llave de tipo $c[j]$ previamente. Nótese que el jugador **nunca** descarta una llave que haya recogido.

El jugador **empieza** el juego en un cuarto s sin haber recogido ninguna llave. Un cuarto t es **alcanzable** desde un cuarto s, si el jugador que empieza el juego en el cuarto s puede realizar una secuencia de acciones, y alcanzar el cuarto t.

Para cada cuarto i ($0 \le i \le n-1$), el número de cuartos alcanzables desde el cuarto i se denota como $p[i]$. Timothy quiere saber el conjunto de índices i que obtienen el mínimo valor de $p[i]$ para $0 \le i \le n-1$.

Detalles de implementación

Se debe implementar la siguiente función:

```
int[] find_reachable(int[] r, int[] u, int[] v, int[] c)
```

- r: es un arreglo de tamaño n. Para cada i ($0 \le i \le n-1$), la llave en el cuarto i es de tipo $r[i]$.
- u, v: dos arreglos de tamaño m. Para cada j ($0 \le j \le m-1$), el conector j conecta los cuartos $u[j]$ y $v[j]$.

- c: un arreglo de tamaño m. Para cada j ($0 \leq j \leq m-1$), el tipo de llave necesaria para recorrer el conector j es $c[j]$.
- Esta función debe retornar un arreglo a de tamaño n. Para cada $0 \leq i \leq n-1$, el valor de $a[i]$ debe ser 1 si para cada j tal que $0 \leq j \leq n-1$, $p[i] \leq p[j]$. De lo contrario, el valor de $a[i]$ debe ser 0.

Ejemplos

Ejemplo 1

Considere la siguiente llamada:

```
find_reachable([0, 1, 1, 2],
               [0, 0, 1, 1, 3], [1, 2, 2, 3, 1], [0, 0, 1, 0, 2])
```

Si el jugador empieza el juego en el cuarto 0, puede realizar la siguiente secuencia de acciones:

Cuarto actual	Acción
0	Recoger llave de tipo 0
0	Recorrer conector 0 hacia cuarto 1
1	Recoger llave de tipo 1
1	Recorrer conector 2 hacia cuarto 2
2	Recorrer conector 2 hacia cuarto 1
1	Recorrer conector 3 hacia cuarto 3

De modo que el cuarto 3 es alcanzable desde el cuarto 0. Similarmente, podemos construir secuencias que muestren que todos los cuartos son alcanzables desde el cuarto 0, lo que implica que $p[0] = 4$. La tabla inferior muestra los cuartos alcanzables para todos los cuartos iniciales:

Cuarto inicial i	Cuartos alcanzables	$p[i]$
0	$[0, 1, 2, 3]$	4
1	$[1, 2]$	2
2	$[1, 2]$	2
3	$[1, 2, 3]$	3

El mínimo valor de $p[i]$ entre todos los cuartos es 2, el cual se obtiene para $i = 1$ o $i = 2$. Por lo tanto, la función debe retornar $[0, 1, 1, 0]$.

Ejemplo 2

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

La tabla inferior muestra los cuartos alcanzables:

Cuarto inicial i	**Cuartos alcanzables**	$p[i]$
0	$[0, 1, 2, 3, 4, 5, 6]$	7
1	$[1, 2]$	2
2	$[1, 2]$	2
3	$[3, 4, 5, 6]$	4
4	$[4, 6]$	2
5	$[3, 4, 5, 6]$	4
6	$[4, 6]$	2

El mínimo valor de $p[i]$ entre todos los cuartos es 2, el cual se obtiene para $i \in \{1, 2, 4, 6\}$. Por lo tanto, la función debe retornar $[0, 1, 1, 0, 1, 0, 1]$.

Ejemplo 3

```
find_reachable([0, 0, 0], [0], [1], [0])
```

La tabla inferior muestra los cuartos alcanzables:

Cuarto inicial i	**Cuartos alcanzables**	$p[i]$
0	$[0, 1]$	2
1	$[0, 1]$	2
2	$[2]$	1

El mínimo valor de $p[i]$ entre todos los cuartos es 1, el cual se obtiene para $i = 2$. Por lo tanto, la función debe retornar $[0, 0, 1]$.

Restricciones

- $2 \le n \le 300\,000$
- $1 \le m \le 300\,000$
- $0 \le r[i] \le n - 1$ para todo $0 \le i \le n - 1$
- $0 \le u[j], v[j] \le n - 1$ y $u[j] \neq v[j]$ para todo $0 \le j \le m - 1$

- $0 \leq c[j] \leq n-1$ para todo $0 \leq j \leq m-1$

Subtareas

1. (9 puntos) $c[j]=0$ para todo $0 \leq j \leq m-1$ y $n, m \leq 200$
2. (11 puntos) $n, m \leq 200$
3. (17 puntos) $n, m \leq 2000$
4. (30 puntos) $c[j] \leq 29$ (para todo $0 \leq j \leq m-1$) y $r[i] \leq 29$ (para todo $0 \leq i \leq n-1$)
5. (33 puntos) Sin restricciones adicionales.

Evaluador de ejemplo

El evaluador de ejemplo lee la entrada en el siguiente formato:

- línea 1: $n \;\; m$
- línea 2: $r[0] \;\; r[1] \;\; \ldots \;\; r[n-1]$
- línea $3+j$ ($0 \leq j \leq m-1$): $u[j] \;\; v[j] \;\; c[j]$

El evaluador de ejemplo imprime el valor retornado por `find_reachable` en el siguiente formato:

- línea 1: $a[0] \;\; a[1] \;\; \ldots \;\; a[n-1]$