

Llaves

Timothy el arquitecto ha diseñado un nuevo juego de escape. En este juego, hay n habitaciones numeradas de 0 a $n - 1$. Inicialmente, cada habitación contiene exactamente una llave. Cada llave tiene un tipo, que es un número entero entre 0 y $n - 1$, ambos inclusive. El tipo de llave de la habitación i ($0 \leq i \leq n - 1$) es $r[i]$. Tenga en cuenta que varias habitaciones pueden contener llaves del mismo tipo, es decir, los valores $r[i]$ no son necesariamente distintos.

También hay m conectores **bidireccionales** en el juego, numerados de 0 a $m - 1$. El conector j ($0 \leq j \leq m - 1$) conecta un par de habitaciones diferentes $u[j]$ y $v[j]$. Un par de habitaciones puede estar conectado por múltiples conectores.

El juego es jugado por un solo jugador que recoge las llaves y se mueve entre las habitaciones atravesando los conectores. Decimos que el jugador **atraviesa** el conector j cuando utiliza este conector para pasar de la habitación $u[j]$ a la habitación $v[j]$, o viceversa. El jugador sólo puede atravesar el conector j si antes ha recogido una llave del tipo $c[j]$.

En cualquier punto durante el juego, el jugador se encuentra en una habitación x y puede realizar dos tipos de acciones:

- Recoger la llave en la sala x , cuyo tipo es $r[x]$ (a menos que ya la haya recogido),
- Atravesar un conector j , en el que $u[j] = x$ o $v[j] = x$, si el jugador ha recogido previamente una llave de tipo $c[j]$. Tenga en cuenta que el jugador **nunca** descarta una llave que haya recogido.

El jugador **comienza** el juego en alguna sala s sin llevar ninguna llave. Una sala t es **alcanzable** desde una sala s , si el jugador que comienza el juego en la sala s puede realizar alguna secuencia de acciones descritas anteriormente, y llegar a la sala t .

Para cada habitación i ($0 \leq i \leq n - 1$), denote el número de habitaciones alcanzables desde la habitación i como $p[i]$. Timothy desea conocer el conjunto de índices i que alcanzan el valor mínimo de $p[i]$ a través de $0 \leq i \leq n - 1$.

Detalles de la implementación

Debe aplicar el siguiente procedimiento:

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

- r : una matriz de longitud n . Para cada i ($0 \leq i \leq n - 1$), la llave de la sala i es de tipo $r[i]$.

- u, v : dos matrices de longitud m . Para cada j ($0 \leq j \leq m - 1$), el conector j conecta las habitaciones $u[j]$ y $v[j]$.
- c : una matriz de longitud m . Para cada j ($0 \leq j \leq m - 1$), el tipo de llave necesaria para recorrer el conector j es $c[j]$.
- Este procedimiento debe devolver una matriz a de longitud n . Para cada $0 \leq i \leq n - 1$, el valor de $a[i]$ debe ser 1 si para cada j tal que $0 \leq j \leq n - 1$, $p[i] \leq p[j]$. En caso 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 comienza el juego en la habitación 0, puede realizar la siguiente secuencia de acciones:

Sala actual	Acción
0	Recoger la llave de tipo 0
0	Atravesar el conector 0 a la habitación 1
1	Recoger la llave de tipo 1
1	Recorrer el conector 2 a la habitación 2
2	Recorrer el conector 2 a la habitación 1
1	Recorrer el conector 3 a la habitación 3

1 | *Recoger la llave de tipo 0* | Recorrer el conector de 2 a la habitación de 2 | Recorrer el conector de 2\$ a la habitación de 1\$. | Recorrer el conector de 3\$ a la habitación de 3\$

Por tanto, la habitación 3 es alcanzable desde la habitación 0. Del mismo modo, podemos construir secuencias que muestren que todas las habitaciones son alcanzables desde la habitación 0, lo que implica que $p[0] = 4$. La siguiente tabla muestra las habitaciones alcanzables para todas las habitaciones iniciales:

Sala de inicio i	Habitaciones alcanzables	$p[i]$
0	[0, 1, 2, 3]	4
1	[1, 2]	2
2	[1, 2]	2
3	[1, 2, 3]	3

El valor más pequeño de $p[i]$ en todas las habitaciones es 2, y esto se consigue para $i = 1$ o $i = 2$. Por lo tanto, este procedimiento debería devolver [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 siguiente tabla muestra las salas accesibles:

Sala de inicio i	Habitaciones 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 valor más pequeño de $p[i]$ en todas las habitaciones es de 2, y esto se consigue para $i \in \{1, 2, 4, 6\}$. Por lo tanto, este procedimiento debe devolver [0, 1, 1, 0, 1, 0, 1].

Ejemplo 3

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

La siguiente tabla muestra las salas accesibles:

Sala de inicio i	Habitaciones alcanzables	$p[i]$
0	$[0, 1]$	2
1	$[0, 1]$	2
2	$[2]$	1

El valor más pequeño de $p[i]$ en todas las habitaciones es 1, y se alcanza cuando $i = 2$. Por lo tanto, este procedimiento debería devolver $[0, 0, 1]$.

Restricciones

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

Subtareas

1. (9 puntos) $c[j] = 0$ para todos los $0 \leq j \leq m - 1$ 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) No hay restricciones adicionales.

Ejemplo del Calificador

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

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

El calificador de ejemplo imprime el valor de retorno de `find_reachable` en el siguiente formato:

- línea 1: $a[0] \ a[1] \ \dots \ a[n - 1]$