

# Llaves

Timothy el arquitecto ha diseñado un nuevo juego de escape. En este juego hay  $n$  cuartos numerados del 0 al  $n - 1$ . Inicialmente, cada cuarto contiene exactamente una llave. Cada llave tiene un tipo, el cual es un entero ente 0 y  $n - 1$ , inclusive. El tipo de llave en el cuarto  $i$  ( $0 \leq i \leq n - 1$ ) es  $r[i]$ . Nota que varios cuartos pueden tener el mismo tipo de llave, i.e. los valores  $r[i]$  no son necesariamente distintos.

Tambien hay  $m$  conexiones **bidireccionales** en el juego, numeradas del 0 al  $m - 1$ . La conexión  $j$  ( $0 \leq j \leq m - 1$ ) conecta un par de cuartos diferentes  $u[j]$  y  $v[j]$ . Un par de cuartos pueden estar conectados por múltiples conexiones.

El juego es jugado por un solo jugador que recoge llaves y se mueve por los cuartos recorriendo las conexiones. Decimos que el jugador **recorre** la conexión  $j$  cuando usan la conexión para moverse del cuarto  $u[j]$  al cuarto  $v[j]$ , o viceversa. El jugador solo puede recorrer una conexión  $j$  si han recogido una llave de tipo  $c[j]$  antes.

En cualquier punto durante el juego, el jugador esta en un cuarto  $x$  y puede realizar dos tipos de acciones:

- Recoger la llave en el cuarto  $x$ , cuyo tipo es  $r[x]$  (a menos que ya la hayan recogido antes),
- Recorrer la conexión  $j$ , donde  $u[j] = x$  o  $v[j] = x$ , si el jugador ha recogido una llave del tipo  $c[j]$  anteriormente. Nota que el jugador **nunca** descarta una llave que ha recogido.

El jugador **inicia** en algun cuarto  $s$  sin llevar ninguna llave. El cuarto  $t$  es **alcanzable** desde un cuarto  $s$ , si el jugador empieza en el cuarto  $s$  y puede realizar una sequencia de las acciones descritas arriba, y alcanzar el cuarto  $t$ .

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

## Detalles de implementación

Debes implementar el siguiente procedimiento:

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

- $r$ : un arreglo de tamaño  $n$ . Para cada  $i$  ( $0 \leq i \leq n - 1$ ), la llave en el cuarto  $i$  es del tipo  $r[i]$ .
- $u, v$ : dos arreglos de tamaño  $m$ . Para cada  $j$  ( $0 \leq j \leq m - 1$ ), la conexión  $j$  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 la conexión  $j$  es  $c[j]$ .
- El procedimiento debe regresar 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 otra manera el valor de  $a[i]$  debe ser 0.

## Ejemplos

### Ejemplo 1

Considera 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 conexión 0 al cuarto 1
1	Recoger llave de tipo 1
1	Recorrer conexión 2 al cuarto 2
2	Recorrer conexión 2 al cuarto 1
1	Recorrer conexión 3 al cuarto 3

Por lo tanto el cuarto 3 es alcanzable desde el cuarto 0. Semejantemente podemos construir una secuencia mostrando que todos los cuartos son alcanzables desde el cuarto 0, que implica que  $p[0] = 4$ . La tabla de abajo muestra los cuartos alcanzables desde cada uno de 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 valor mas pequeño de  $p[i]$  a través de todos los cuartos es 2, y es logrado por  $i = 1$  y  $i = 2$ . Por lo tanto el procedimiento debe regresar [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 de abajo 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 menor valor de  $p[i]$  a través de todos los cuartos es 2, y es logrado por  $i \in \{1, 2, 4, 6\}$ . Por lo tanto, el procedimiento debe regresar [0, 1, 1, 0, 1, 0, 1].

### Ejemplo 3

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

La tabla de abajo muestra los cuartos alcanzables:

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

El menor valor de  $p[i]$  a través de todos los cuartos es 1, y es alcanzado cuando  $i = 2$ . Por lo tanto el procedimiento debe regresar [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 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 condiciones adicionales.

## Evaluador de ejemplo

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

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

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

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