

CS2023 - Aula de Ejercicios N° 9  
**Brenner H. Ojeda Rios**  
Semestre 2024-1

Se sugiere que cada estudiante trate de resolver los ejercicios de forma **individual** y luego los discuta en grupo.

## Ejercicios

1. (6 pts) En un pueblo, hay  $n$  personas etiquetadas del 1 al  $n$ . Se rumorea que una de estas personas es secretamente el juez del pueblo.

Si el juez del pueblo existe, entonces:

- El juez del pueblo no confía en nadie.
- Todos (excepto el juez del pueblo) confían en el juez del pueblo.
- Hay exactamente una persona que satisface las propiedades 1 y 2.

Se te da un arreglo de confianza (**trust**) donde  $\text{trust}[i] = [a_i, b_i]$  representa que la persona etiquetada como  $a_i$  confía en la persona etiquetada como  $b_i$ . Si una relación de confianza no existe en el arreglo de confianza, entonces tal relación de confianza no existe.

Devuelve la etiqueta del juez del pueblo si el juez del pueblo existe y puede ser identificado, o devuelve -1 en caso contrario.

### Ejemplo 1:

Entrada:  $n = 2$ ,  $\text{trust} = [[1,2]]$

Salida: 2

### Ejemplo 2:

Entrada:  $n = 3$ ,  $\text{trust} = [[1,3], [2,3]]$

Salida: 3

### Ejemplo 3:

Entrada:  $n = 3$ ,  $\text{trust} = [[1,3], [2,3], [3,1]]$

Salida: -1

### Restricciones:

- $1 \leq n \leq 1000$
- $0 \leq \text{trust.length} \leq 10^4$
- $\text{trust}[i].\text{length} = 2$
- Todos los pares de confianza son únicos.
- $a_i \neq b_i$
- $1 \leq a_i, b_i \leq n$

2. (7 pts) Hay un grafo bidireccional con  $n$  vértices, donde cada vértice está etiquetado del 0 al  $n - 1$  (inclusive). Las aristas en el grafo están representadas como un arreglo bidimensional de enteros `edges`, donde cada `edges[i] = [ui, vi]` denota una arista bidireccional entre el vértice  $u_i$  y el vértice  $v_i$ . Cada par de vértices está conectado por como máximo una arista, y ningún vértice tiene una arista hacia sí mismo.

Se quiere determinar si existe un camino válido desde el vértice `source` hasta el vértice `destination`.

Dados `edges` y los enteros  $n$ , `source` y `destination`, devuelve `true` si existe un camino válido desde `source` hasta `destination`, o `false` en caso contrario.

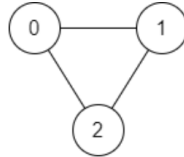
#### Ejemplo 1:

Input:  $n = 3$ , `edges = [[0,1],[1,2],[2,0]]`, `source = 0`, `destination = 2`

Output: `true`

Explanation: Hay dos caminos desde el vértice 0 al vértice 2:

- $0 \rightarrow 1 \rightarrow 2$
- $0 \rightarrow 2$

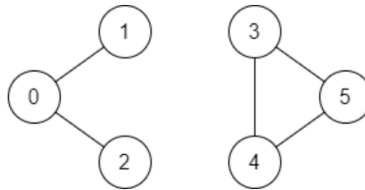


#### Ejemplo 2:

Input:  $n = 6$ , `edges = [[0,1],[0,2],[3,5],[5,4],[4,3]]`, `source = 0`, `destination = 5`

Output: `false`

Explanation: No hay un camino desde el vértice 0 al vértice 5.



#### Restricciones:

- $1 \leq n \leq 2 \times 10^5$
- $0 \leq \text{edges.length} \leq 2 \times 10^5$
- `edges[i].length = 2`
- $0 \leq u_i, v_i \leq n - 1$
- $u_i \neq v_i$
- $0 \leq \text{source}, \text{destination} \leq n - 1$
- No hay aristas duplicadas.
- No hay aristas hacia sí mismas.

3. (7 pts) Hay un grafo estrella no dirigido que consiste en  $n$  nodos etiquetados del 1 al  $n$ . Un grafo estrella es un grafo donde hay un nodo central y exactamente  $n - 1$  aristas que conectan el nodo central con cada uno de los otros nodos.

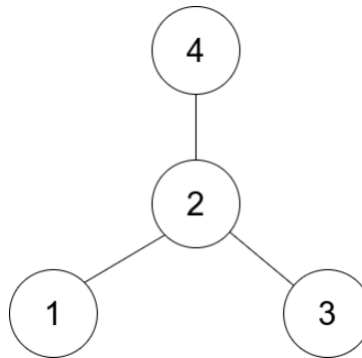
Se te da un arreglo bidimensional de enteros `edges`, donde cada `edges[i] = [ui, vi]` indica que hay una arista entre los nodos  $u_i$  y  $v_i$ . Devuelve el centro del grafo estrella dado.

**Ejemplo 1:**

Input: `edges = [[1,2],[2,3],[4,2]]`

Output: 2

Explanation: Como se muestra en la figura anterior, el nodo 2 está conectado a todos los otros nodos, por lo que 2 es el centro.



**Ejemplo 2:**

Input: `edges = [[1,2],[5,1],[1,3],[1,4]]`

Output: 1

**Restricciones:**

- $3 \leq n \leq 10^5$
- `edges.length = n - 1`
- `edges[i].length = 2`
- $1 \leq u_i, v_i \leq n$
- $u_i \neq v_i$
- Las aristas dadas representan un grafo estrella válido.