



## Objetivos de la Guía

- Aprender sobre los recorridos de grafos
- Aprender sobre Búsqueda en Profundidad
- Aprender sobre Búsqueda en Anchura
- Conocer sobre las componentes conexas

## > A - La información es lo más importante

Pepito, el tutor que no existe, quiere poder analizar los grafos a través de sus conocimientos aprendidos en OCILabs. Así que te pide una seguidilla de tareas.

### \* Parte 1: ¿Que ciudades hacen parte de la misma componente conexa

La primera parte consiste en leer un grafo no dirigido, en el cual preguntan: ¿Que otras ciudades hacen parte de la misma componente conexa que  $x$ ?

#### \* Input:

La primera línea contiene dos enteros  $n$  y  $m$ , donde  $n$  corresponde a la cantidad de ciudades y  $m$  corresponde a la cantidad de conexiones entre ciudades.

Luego le siguen  $m$  líneas, cada línea contiene dos enteros  $a$  y  $b$  indicando que existe una conexión entre  $a$  y  $b$ . Las ciudades están enumeradas entre 1 y  $n$ .

Finalmente, la última línea contiene un entero  $x$ , que es la ciudad a consultar.

```
6 4
3 2
5 4
6 5
1 2
1
```

#### \* Output:

Muestre por pantalla las ciudades que hacen parte de la misma conexa que la ciudad  $x$ .

```
2 3
```

### Tip 1

Puede utilizar el código de DFS visto en la cápsula.

### Tip 2

Recuerde que el vector de visitados indica si es que los nodos visitados usando DFS o no

### \* Parte 2: Distancias

Ahora, Pepito quiere saber la distancia más corta de ir desde la ciudad  $x$  hasta las otras ciudades del país.

#### \* Input:

La primera línea contiene dos enteros  $n$  y  $m$ , donde  $n$  corresponde a la cantidad de ciudades y  $m$  corresponde a la cantidad de conexiones entre ciudades.

Luego le siguen  $m$  líneas, cada línea contiene dos enteros  $a$  y  $b$  indicando que existe una conexión entre  $a$  y  $b$ . Las ciudades están enumeradas entre 1 y  $n$ .

Finalmente, la última línea contiene un entero  $x$ , que es la ciudad a consultar.



```
6 4
3 2
5 4
6 5
1 2
1
```

## \* Output:

Muestre por pantalla  $n$  números, donde el número en la posición  $i$  corresponde a la distancia más corta de llegar desde  $x$  hasta esa ciudad.

Si la ciudad  $x$  no puede alcanzar la ciudad  $i$ , debe mostrar por pantalla "INF".

```
0 1 2 INF INF INF
```

## Tip 1

Aplique BFS para poder obtener la distancia más corta

## Tip 2

Recuerde que BFS tiene un vector de distancias :)