

# Reporte Examen

Alumno: Kevin Uriel Dulche Jaime

Matricula: 2213026201

## Sección 3: Representación de grafos

### Pregunta b (25 puntos):

Dada una representación de un grafo no dirigido, implemente un algoritmo para recorrer el grafo en anchura (BFS). El algoritmo debe recorrer todos los nodos del grafo y marcarlos como visitados. La implementación debe indicar si el grafo es conexo.

El programa implementa un algoritmo para recorrer un grafo no dirigido en anchura (BFS) y determinar si el grafo es conexo. A continuación, se describe cómo funciona:

- **Función bfs:** Esta función recibe como parámetro la matriz de adyacencia que representa el grafo. Inicia marcando todos los nodos como no visitados. Luego, se inicia el recorrido desde el primer nodo (índice 0) agregándolo a una cola (**queue**) y marcándolo como visitado. Se continúa recorriendo los nodos adyacentes de forma iterativa, marcándolos como visitados y agregándolos a la cola si aún no han sido visitados y si hay una conexión con el nodo actual. Este proceso se repite hasta que no queden nodos por visitar en la cola.
- **Función main:** En la función principal, se define una matriz de adyacencia que representa el grafo. Luego se llama a la función **bfs** para determinar si el grafo es conexo o no, y se imprime el resultado.

El algoritmo BFS recorre el grafo desde un nodo inicial de manera sistemática, explorando todos los nodos a una distancia dada antes de pasar a los nodos más lejanos. Si todos los nodos son visitados durante este recorrido, el grafo se considera conexo; de lo contrario, no lo es.

#### Sección 4: Problemas de distancias más cortas

##### Pregunta b (25 puntos):

Un trágico accidente ocurrió en el Segundo Piso del Periférico, por lo que una ambulancia llegó de inmediato y el paciente debe ser internado en algún hospital lo antes posible. Hay  $N$  ciudades numeradas del 1, 2, 3...  $N$  en la ubicación cercana y al paciente le quedan  $M$  minutos antes de morir, y solo hay un hospital que está situado en  $Y$  y ahora usted está en  $X$ , por lo que hay que acudir al hospital lo antes posible. Debe determinar si el paciente puede salvarse en  $M$  minutos o no.

```
10 7 19 5 7
3 10 8
1 2 9
2 3 3
5 10 3
1 3 7
1 7 1
3 6 2
2 5 6
2 4 8
7 8 1
1 10 6
2 7 1
2 8 2
3 5 2
2 6 5
4 7 4
6 9 9
8 10 8
7 10 2
```

La primera línea contiene cinco números enteros separados por espacios  $N$ ,  $M$ ,  $R$ ,  $X$  e  $Y$  que indican el número de ciudades, al paciente le quedan minutos antes de morir, el número de ruta para llegar al hospital, la posición en la que se encuentra y la ubicación del hospital.

El programa aborda el problema de determinar si un paciente puede ser trasladado a un hospital en un tiempo limitado, dado un conjunto de ciudades y rutas entre ellas. A continuación, se describe cómo funciona:

- **Inicialización de variables:** Se definen las variables que representan el número de ciudades, el tiempo restante del paciente antes de morir, el número de rutas disponibles, la posición inicial del paciente ( $X$ ) y la ubicación del hospital ( $Y$ ).
- **Creación del grafo:** Se crea una matriz de adyacencia llamada grafo para representar las conexiones entre las ciudades. Inicialmente, todas las conexiones se establecen en cero.

- **Agregar rutas al grafo:** Se leen las rutas disponibles y se agregan al grafo, estableciendo los pesos de las conexiones entre las ciudades según el tiempo que se tarda en viajar de una a otra.
- **Algoritmo de Dijkstra:** Se aplica el algoritmo de Dijkstra para encontrar la distancia más corta desde la posición inicial del paciente (X) a todas las demás ciudades. Esto se hace considerando el tiempo restante del paciente como un límite superior para evitar que el paciente muera durante el traslado.
- **Verificación de la posibilidad de salvar al paciente:** Se comprueba si la distancia más corta desde la posición inicial del paciente (X) hasta el hospital (Y) es menor o igual al tiempo restante del paciente. Si es así, se imprime un mensaje indicando que el paciente puede salvarse; de lo contrario, se imprime un mensaje indicando que el paciente no puede salvarse.

El código proporcionado aborda de manera efectiva el problema planteado al utilizar el algoritmo de Dijkstra para encontrar la distancia más corta en un grafo ponderado y luego determinar si el paciente puede ser trasladado al hospital dentro del tiempo límite.

Al modificar el punto de inicio y/o el final y/o los minutos de vida, se puede ver que si se sobrepasa del tiempo de vida, el paciente, ya no sobrevive.