

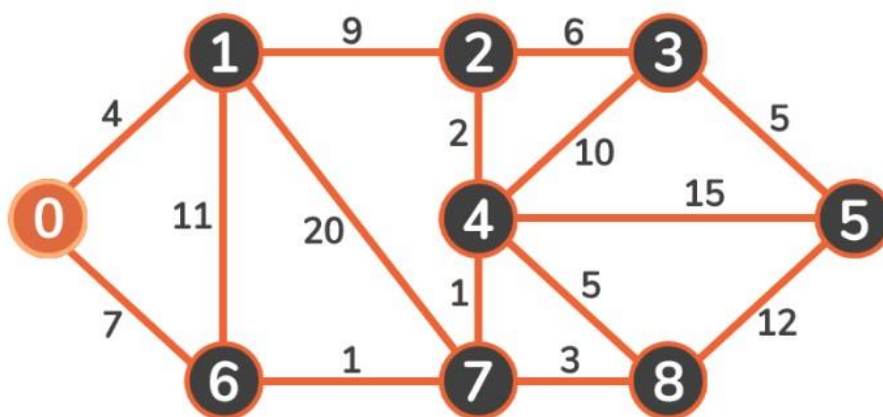
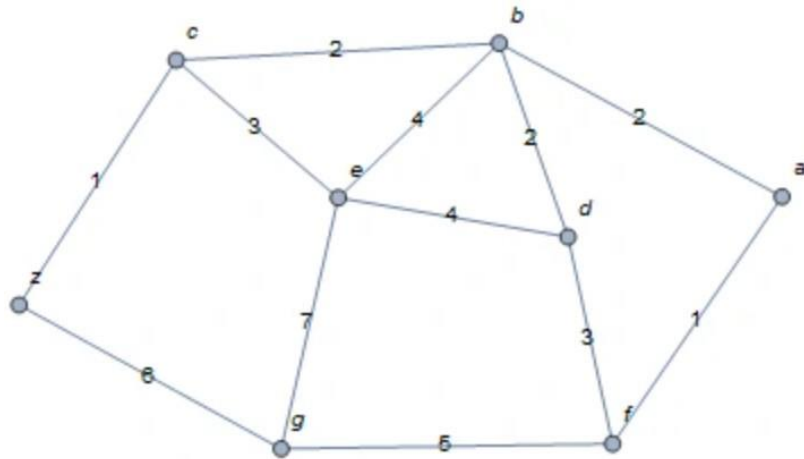
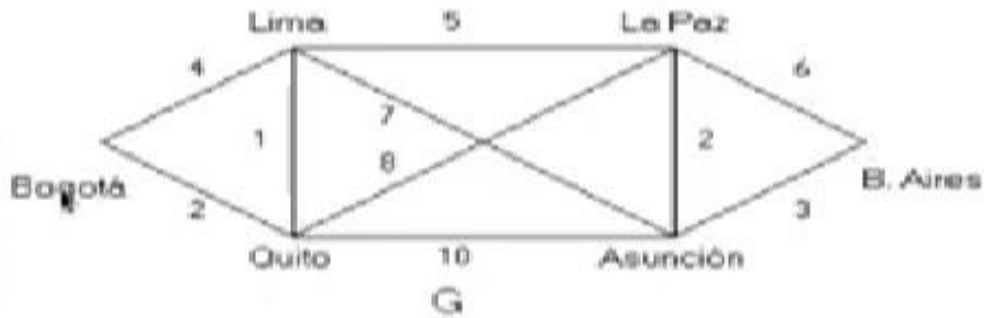


## MATEMATICAS DISCRETAS-PARCIAL III

1. ¿Qué es un grafo ponderado?
2. ¿Qué es un nodo o vértice en un Grafo?
3. ¿Qué son las aristas en un Grafo?
4. ¿Qué son los pesos en un grafo?
5. ¿Cuáles son las formas que existen para representar un grafo?, explique cada una de ellas.
6. ¿Para qué sirve la implementación de la librería **networkx** en un programa de Python?
7. ¿Cuál es la diferencia entre usar el comando **Graph()** y usar el comando **DiGraph()**?
8. ¿Qué es el algoritmo BFS al recorrer un grafo?, dibuje un ejemplo donde se explique su funcionamiento
9. ¿Qué es el algoritmo DFS al recorrer un grafo?, dibuje un ejemplo donde se explique su funcionamiento
10. ¿Cuál de los siguientes tipos de grafos representa mejor una red de transporte público, donde las rutas pueden tener direcciones específicas y distancias asociadas?
  - a) Grafo no dirigido ponderado.
  - b) Grafo dirigido no ponderado.
  - c) Grafo dirigido ponderado.
  - d) Grafo no dirigido no ponderado.
11. ¿Cuál es la función de recorrer un grafo utilizando el algoritmo de Dijkstra?
12. Dado los siguientes grafos: **Realice el proceso de forma manual:**

## MATEMATICAS DISCRETAS-PARCIAL III

Aplicar búsquedas de algoritmos por BFS, DFS, Dijkstra, mostrar las rutas que se siguen para cada uno de los algoritmos.





## MATEMATICAS DISCRETAS-PARCIAL III

### PARCIAL PRACTICO:

Un equipo de ingenieros que está desarrollando un sistema de transporte inteligente para una ciudad, una Red de Transporte Inteligente. El sistema incluye una red de transporte público, que consta de estaciones de autobuses y estaciones de tren interconectadas.

El objetivo es desarrollar un programa que pueda encontrar la ruta más eficiente para un usuario desde una estación de inicio hasta una estación de destino, teniendo en cuenta los diferentes modos de transporte (autobús o tren) y los tiempos de viaje asociados.

Requerimientos:

1. Implementar un grafo dirigido y ponderado que represente la red de transporte público de la ciudad. Cada nodo representa una estación y las aristas representan las conexiones entre las estaciones, con pesos que representan los tiempos de viaje.
2. Desarrollar una función o método que utilice el algoritmo DFS para encontrar todas las posibles rutas desde una estación de inicio hasta una estación de destino. La función debe mostrar las rutas encontradas, teniendo en cuenta las restricciones de tiempo.
3. Desarrollar una función o método que utilice el algoritmo BFS para encontrar la ruta más corta desde una estación de inicio hasta una estación de destino. La función debe mostrar la ruta más corta y el tiempo total de viaje.
4. Implementa el algoritmo de Dijkstra para encontrar la ruta más corta desde una estación de inicio hasta una estación de destino. La función debe mostrar la ruta más corta y el tiempo total de viaje.

Se tiene la siguiente red de transporte público:

- Estaciones: A, B, C, D, E, F, G
- Conexiones:



## MATEMATICAS DISCRETAS-PARCIAL III

- A  $\rightarrow$  B (tiempo: 5 min, tren)
- A  $\rightarrow$  C (tiempo: 10 min, autobús)
- B  $\rightarrow$  D (tiempo: 15 min, autobús)
- C  $\rightarrow$  D (tiempo: 20 min, tren)
- C  $\rightarrow$  E (tiempo: 5 min, autobús)
- D  $\rightarrow$  F (tiempo: 10 min, tren)
- E  $\rightarrow$  F (tiempo: 15 min, tren)
- E  $\rightarrow$  G (tiempo: 5 min, autobús)
- F  $\rightarrow$  G (tiempo: 10 min, autobús)