PARCIAL 3

import matplotlib.pyplot  as plt

import networkx as nx

nuevo\_grafo = nx.Graph();

nuevo\_grafo.add\_nodes\_from(["A","B","C","D","E","F","G"]);

nuevo\_grafo.add\_edge("A","B",weight = 5);

nuevo\_grafo.add\_edge("A","C",weight = 10);

nuevo\_grafo.add\_edge("B","D",weight = 15);

nuevo\_grafo.add\_edge("C","D",weight = 20);

nuevo\_grafo.add\_edge("C","E",weight = 5);

nuevo\_grafo.add\_edge("D","F",weight = 10);

nuevo\_grafo.add\_edge("E","F",weight = 15);

nuevo\_grafo.add\_edge("E","G",weight = 5);

nuevo\_grafo.add\_edge("E","G",weight = 10);

nx.posicion = nx.spring\_layout(nuevo\_grafo);

Estacion\_Inicial= str(input("Digite una letra mayuscula entre la A y la G que representan las estaciones de transporte publico que va a ser la estacion inicial: "));

Estacion\_Final= str(input("Digite una letra mayuscula entre la A y la G que representan las estaciones de transporte publico que va a ser la estacion final: "))

# Mostrar recorrido BFS

print(" Posibles rutas con el recorrido BFS");

bfs\_edges = nx.bfs\_edges(nuevo\_grafo,source=Estacion\_Inicial);

print(list(bfs\_edges));

# Mostrar recorrido DFS

print("Posibles rutas con el recorrido DFS");

dfs\_edges = nx.dfs\_edges(nuevo\_grafo,source=Estacion\_Final);

print(list(dfs\_edges));

nx.draw\_networkx(nuevo\_grafo,nx.posicion,with\_labels = True, node\_color = 'lightblue',font\_weight = 'bold');

labels = nx.get\_edge\_attributes(nuevo\_grafo,'weight');

nx.draw\_networkx\_edge\_labels(nuevo\_grafo, nx.posicion, edge\_labels = labels);

plt.show();

# Algoritmo dijkstra

Inicio,Final = input("Ingrese las estaciones de inicio y final separados por espacios = ").split();

path = nx.dijkstra\_path(nuevo\_grafo, source = Inicio , target = Final);

print("Ruta mas corta = ".format(Inicio,Final),path);

distance = nx.dijkstra\_path\_length(nuevo\_grafo, source=Inicio, target= Final);

print(" La distancia de la ruta mas corta es de = ".format(Inicio,Final),distance);