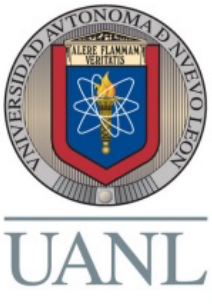
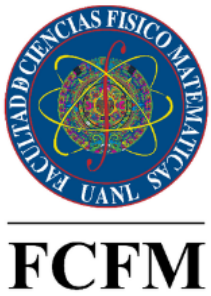


Universidad Autónoma De Nuevo León
Facultad de Ciencias Físico Matemáticas



Alumnos:

Leonardo Martin Vázquez Cruz 1679807
Bryan Aguirre Tudon
Alexis Emiliano Eguía Sifuentes

Docente

Juan Pablo Rosas Baldazo

Materia: Inteligencia Artificial

Tarea: 2

Instrucciones:

De acuerdo con lo visto en la clase anterior, deben de programar un generador de instancias que sea capaz de crear instancias de grafos con las siguientes características

- Un mínimo de 20 nodos, de preferencia que sean más de 40, ya que para problemas como ruta más corta es más interesante
- Que sea capaz de crear grafos dirigidos o no
- Que los arcos puedan tener peso
- Como extra pueden considerar que su generador de instancias pueda crear grafos completos o no

```python
# Una clase para representar un objeto grafo
class Grafo:
    # Constructor para construir un grafo
    def __init__(self, bordes, n):

        # Una lista de listas para representar una lista de adyacencia
        self.adjList = [None] * n

        # asigna memoria para la lista de adyacencia
        for i in range(n):
            self.adjList[i] = []

        # agrega bordes al grafo dirigido
        for (src, dest, weight) in bordes:
            # asignar nodo en la lista de adyacencia de src a dest
            self.adjList[src].append((dest, weight))


# Función para imprimir la representación de lista de adyacencia de un grafo
def printG(grafo):
    for src in range(len(grafo.adjList)):
        # imprime el vértice actual y todos sus vértices vecinos
        for (dest, weight) in grafo.adjList[src]:
            print(f'({src} -> {dest}, {weight}) ', end='')
        print()
```

```python
if __name__ == '__main__':

    # Entrada: bordes en un dígrafo ponderado (según el diagrama anterior)
    # Arista (x, y, w) representa una arista de `x` a `y` con peso `w`
    bordes = [(0, 1, 6), (1, 2, 7), (2, 0, 5), (2, 1, 4), (3, 2, 10),
              (4, 5, 1), (5, 4, 3)]

    # Nº de vértices
    n = 19

    # construye un grafo a partir de una lista dada de aristas
    Grafo = Grafo(bordes, n)

    # imprime la representación de la lista de adyacencia del grafo
    printG(Grafo)
```

```
/ms-python.python-2022.16.1/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 50633 -- /Users/leonardovazquez/Desktop/Inteligencia\ Artific
ial/Traea2.py
(0 -> 1, 6)
(1 -> 2, 7)
(2 -> 0, 5) (2 -> 1, 4)
(3 -> 2, 10)
(4 -> 5, 1)
(5 -> 4, 3)
```

Resultados:

Se utilizaron menos valores, pero si funciona para 20 nodos

```
/ms-python.python-2022.16.1/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 50633 -- /Users/leonardovazquez/Desktop/Inteligencia\ Artific
ial/Traea2.py
(0 -> 1, 6)
(1 -> 2, 7)
(2 -> 0, 5) (2 -> 1, 4)
(3 -> 2, 10)
(4 -> 5, 1)
(5 -> 4, 3)
```