



UNIVERSIDAD DE GUADALAJARA
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

Act. Limitaciones de los algoritmos - Problema del viajero
Análisis de Algoritmos - Mtro. Jorge Ernesto Lopez Arce Delgado

Realizado por: Solís Regin Juan Pablo
Código 220468416

Introducción

Por medio de esta actividad se desea resolver el problema del viajero en Python, usando de ejemplo matrices para representar las ciudades y las rutas que se debe tomar para elegir la ruta de menor costo y la más óptima.

Código en Python:

Python

```
#Programador: Juan Pablo Solís Regín Código: 220468416

import numpy as np
from python_tsp.exact import solve_tsp_dynamic_programming

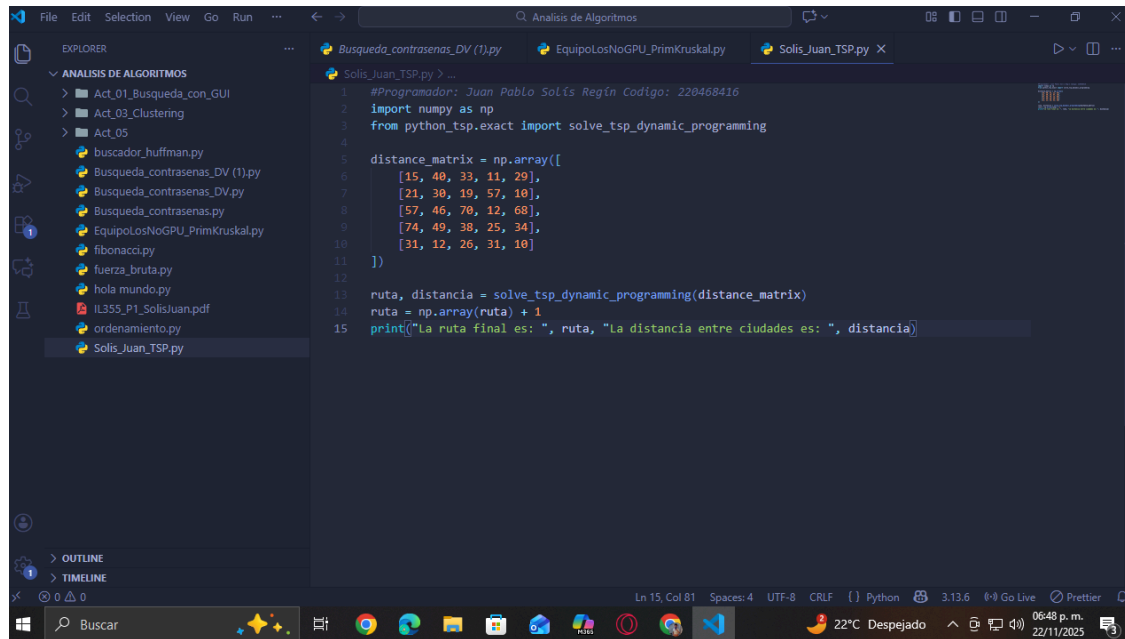
distance_matrix = np.array([
    [15, 40, 33, 11, 29],
    [21, 30, 19, 57, 10],
    [57, 46, 70, 12, 68],
    [74, 49, 38, 25, 34],
    [31, 12, 26, 31, 10]
])

ruta, distancia = solve_tsp_dynamic_programming(distance_matrix)
ruta = np.array(ruta) + 1
print("La ruta final es: ", ruta, "La distancia entre ciudades es: ", distancia)
```

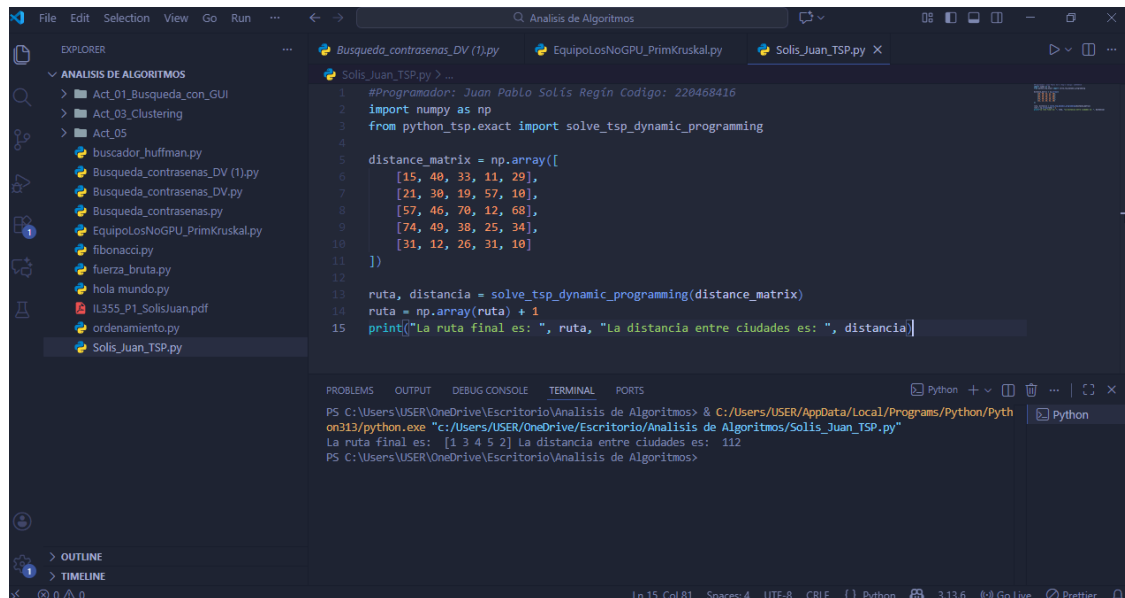
Explicacion del codigo:

1. Se hace la instalación con el comando `pip install python-tsp` para tener la biblioteca de Python `python-tsp`, que vamos a utilizar para el problema del viajero.
2. `from python_tsp.exact import solve_tsp_dynamic_programming` se utiliza para resolver el problema con programación dinámica.
3. La matriz que se utiliza es de 5x5, lo que simula 5 ciudades con sus pesos de distancias para ir de un lugar a otro.
4. `ruta, distancia = solve_tsp_dynamic_programming(distance_matrix)` implementa un algoritmo exacto que garantiza encontrar la solución óptima.
5. Al final se utiliza un `print` para imprimir los resultados sobre la ruta final y la distancia recorrida final.

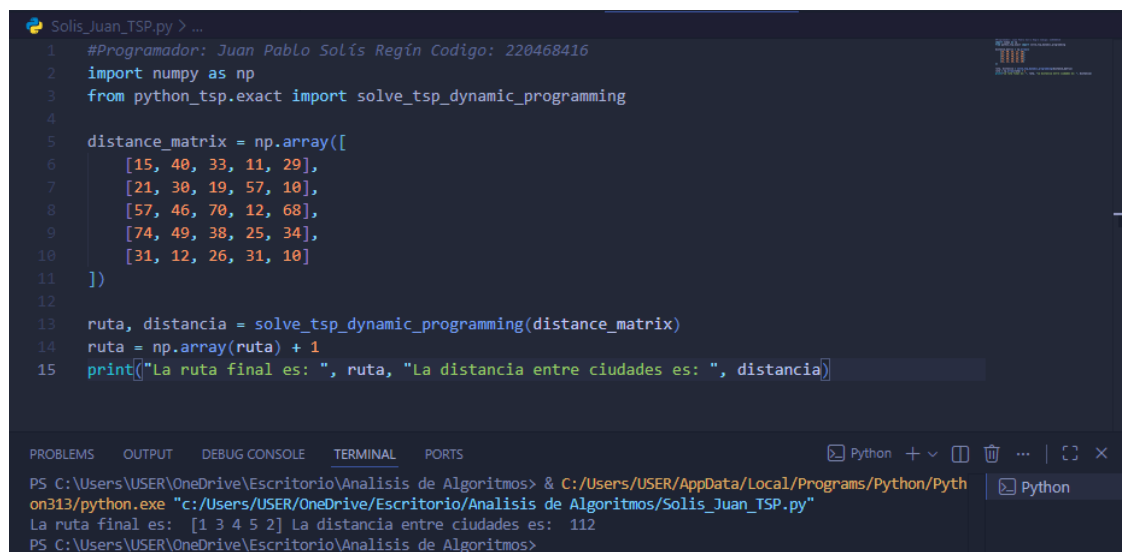
Capturas de pantalla:



```
1 #Programador: Juan Pablo Solís Regín Código: 220468416
2 import numpy as np
3 from python_tsp.exact import solve_tsp_dynamic_programming
4
5 distance_matrix = np.array([
6     [15, 40, 33, 11, 29],
7     [21, 30, 19, 57, 10],
8     [57, 46, 70, 12, 68],
9     [74, 49, 38, 25, 34],
10    [31, 12, 26, 31, 10]
11 ])
12
13 ruta, distancia = solve_tsp_dynamic_programming(distance_matrix)
14 ruta = np.array(ruta) + 1
15 print("La ruta final es: ", ruta, "La distancia entre ciudades es: ", distancia)
```



```
PS C:\Users\USER\OneDrive\Escritorio\Análisis de Algoritmos> & C:/Users/USER/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/USER/OneDrive/Escritorio/Análisis de Algoritmos/Solis_Juan_TSP.py"
La ruta final es: [1 3 4 5 2] La distancia entre ciudades es: 112
PS C:\Users\USER\OneDrive\Escritorio\Análisis de Algoritmos>
```



```
1 #Programador: Juan Pablo Solís Regín Código: 220468416
2 import numpy as np
3 from python_tsp.exact import solve_tsp_dynamic_programming
4
5 distance_matrix = np.array([
6     [15, 40, 33, 11, 29],
7     [21, 30, 19, 57, 10],
8     [57, 46, 70, 12, 68],
9     [74, 49, 38, 25, 34],
10    [31, 12, 26, 31, 10]
11 ])
12
13 ruta, distancia = solve_tsp_dynamic_programming(distance_matrix)
14 ruta = np.array(ruta) + 1
15 print("La ruta final es: ", ruta, "La distancia entre ciudades es: ", distancia)
```

```
PS C:\Users\USER\OneDrive\Escritorio\Análisis de Algoritmos> & C:/Users/USER/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/USER/OneDrive/Escritorio/Análisis de Algoritmos/Solis_Juan_TSP.py"
La ruta final es: [1 3 4 5 2] La distancia entre ciudades es: 112
PS C:\Users\USER\OneDrive\Escritorio\Análisis de Algoritmos>
```