

UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

INGENIERO EN COMPUTACIÓN

ANÁLISIS DE ALGORITMOS

MACIEL VARGAS OSWALDO DANIEL

GARCÍA SALDIVAR HUGO GABRIEL

Presentación #01

Fuerza Bruta

I. Roles y Responsabilidades:

Oswaldo:

- Elegir el algoritmo
- Probar el un prototipo del algoritmo elegido en vs code
- Creación de la plantilla para la presentación

Hugo:

- Buscar aplicaciones para el algoritmo elegido
- Desarrollar el algoritmo a profundidad

II. Algoritmo

Generación de cuadrados latinos

Elegimos este algoritmo ya que encontramos muchas aplicaciones, desde las más comunes hasta las más complejas. Es fácil de comprender ya que lo que se busca es que no haya elementos repetidos ni en filas ni columnas, lo que nos permite diseñar experimentos donde no se repitan casos o elementos.

III. Problema de viajero

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "ALGORITMO PROBLEMA DEL VIAJERO GRAFOS" containing a file "main.py".
- Code Editor:** Displays the following Python code for finding the optimal route in a graph:

```
File Edit Selection View Go Run ... Algoritmo Problema del Viajero Grafos
EXPLORER main.py x
ALGORITMO PROBLEMA DEL VIAJERO GRAFOS main.py
18 def encontrar_ruta_optima_grafo(grafo):
19     for perm in permutaciones:
20         ruta_actual = [punto_inicio] + list(perm)
21         distancia_actual = distancia_ruta(ruta_actual, grafo)
22
23         if distancia_actual < minima_distancia:
24             minima_distancia = distancia_actual
25             mejor_ruta = ruta_actual
26
27         mejor_ruta.append(punto_inicio)
28
29     return mejor_ruta, minima_distancia
30
31
32 grafo_ciudades = {
33     'A': {'B': 12, 'C': 11, 'D': 18, 'E': 20},
34     'B': {'A': 12, 'C': 25, 'D': 5, 'E': 9},
35     'C': {'A': 11, 'B': 25, 'D': 30, 'E': 11},
36     'D': {'A': 18, 'B': 5, 'C': 30, 'E': 9},
37     'E': {'A': 20, 'B': 9, 'C': 11, 'D': 9}
38 }
39
40
41 mejor_ruta, minima_distancia = encontrar_ruta_optima_grafo(grafo_ciudades)
42
43 print("\nLa ruta mas optima encontrada fue:")
44 print(" -> ".join(map(str, mejor_ruta)))
45 print(f"\nCon una distancia total de: {minima_distancia}\n")
46
47
48
49
50
51
52
53
54
```

Terminal: Shows the command line output of the script execution:

```
PS C:\Users\Yogo\Desktop\CUCEI\5to semestre\Analisis Algoritmos\Algoritmo Problema del Viajero Grafos & C:/Users/Yogo/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Yogo/Desktop/CUCEI/5to semestre/Analisis Algoritmos/Algoritmo Problema del Viajero Grafos/main.py"
La ruta mas optima encontrada fue:
A -> B -> D -> E -> C -> A
Con una distancia total de: 48
```

