

CI-2693: Laboratorio de Algoritmos y Estructuras 3

Laboratorio Nr. 4

26 de junio de 2024

Instrucciones Generales

- La solución debe ser subida a un repositorio privado en GitHub e incluir un archivo `README.md` con la complejidad temporal y espacial de cada solución, junto con sus respectivas argumentaciones.
- Deben escribir casos de prueba para cada algoritmo utilizando `unittest` o `pytest`. Consultar la documentación de `unittest`.
- El código debe estar adecuadamente comentado utilizando docstrings y especificar el tipo de entrada y salida en los argumentos de cada función.
- Cada equipo debe estar conformado por 2 a 3 integrantes.

Ejercicio 1

Andrés ama viajar por Venezuela y desea optimizar sus rutas para minimizar el consumo de gasolina, considerando las pendientes variables de las carreteras. Tu tarea consiste en implementar una función que calcule la ruta con el menor consumo de gasolina utilizando el algoritmo de Dijkstra, teniendo en cuenta la distancia y la pendiente entre ciudades.

Escenario

Andrés comienza su viaje en Caracas y desea viajar a Mérida. Las carreteras entre ciudades tienen pendientes variables que afectan el consumo de gasolina

y el tiempo de viaje. Debes encontrar la ruta óptima para Andrés que minimice el consumo de gasolina, respetando la restricción de que el tiempo de viaje puede extenderse en un máximo del `max_time_extension%` en comparación con el tiempo ideal sin preocupaciones por la gasolina. La ruta óptima se refiere a la ruta con el menor tiempo.

Entrada

- **Red de Carreteras:** Un conjunto de ciudades conectadas por carreteras con distancias conocidas en kilómetros.
- **Información de Pendiente:** Cada carretera tiene una pendiente en grados, que va de -90 a 90.
- **Consumo de Gasolina:** Consumo de gasolina que se ajusta $\pm x\%$ por grado.
- **Velocidad Inicial en km/h**
- **Gasolina Inicial en litros**
- **Ciudad inicial**
- **Ciudad destino**

Requerimientos de la Función

Tu función debe calcular la ruta óptima utilizando el algoritmo de Dijkstra, considerando las variables mencionadas.

Implementación

Implementación del Algoritmo de Dijkstra con Cola de Prioridad

Para este ejercicio, implementa tu propia cola de prioridad que actualice en tiempo logarítmico. Consulta referencias como Cormen et al. para implementar esta estructura de datos.

Ejemplo

```
start_city = 'Caracas'
end_city = 'Merida'
road_network = {
```

```

    'Caracas': [( 'Maracay ', 120, 5)],
    'Maracay': [( 'Caracas ', 120, 5), ( 'Valencia ', 80, -2)],
    'Valencia': [( 'Maracay ', 80, -2), ( 'Merida ', 180, 3)],
    'Merida': [( 'Valencia ', 180, 3)]
}

initial_velocity = 40 # km/h
gasoline_consumption_per_km = 0.1 # litros
initial_gasoline = 40 # litros
max_time_extension_factor = 1.2

```

Ejercicio 2

Mismo ejercicio pero ahora con una adición: tienes una lista de lugares donde puedes recargar gasolina gratis, cortesía de un amigo. Debes implementar el mismo tipo de función recibiendo este argumento adicional, pero esta vez utilizando el algoritmo de Bellman-Ford.