

Informe 1 – Fecha máxima de entrega: Día de clase de la semana 5.

Criterios de entrega:

- La entrega se realizará por interactiva mediante el enlace al repositorio (Público) con la respectiva solución. (Recuerden que debe ser evidente el aporte de cada integrante)
- Definir los grupos de trabajo (Max. 3 integrantes)
- Cada ejercicio debe tener un documento (README.md) donde se detalle el análisis realizado y se responda a las preguntas planteadas.

Ejercicio 1

A partir del Notebook “2.BestFirstSearch.ipynb” de la semana 2, resolver el problema que soluciona la ruta óptima hasta Bucharest, mediante el algoritmo de A*Search usando la heurística.

El README.md debe explicar:

- El análisis del problema.
- Cómo se aplica A*
- ¿Por qué se considera que la ruta encontrada es óptima?

Ejercicio 2

Notebook “Maze_problem.ipynb”

1. Resolver el ejercicio planteado.
2. ¿Cómo cambia el comportamiento del algoritmo si cambiamos la función de costo?
3. ¿Qué sucede si hay múltiples salidas en el laberinto? ¿Cómo podrías modificar el algoritmo para manejar esto? Plantea una propuesta.
4. Modifica el laberinto por uno más grande y con otro tipo de obstáculo además de paredes. ¿Qué limitación encuentras en el algoritmo?

Ejercicio 3

Navegación en una Red de Metro

Contexto del Problema: Como desarrollador de la alcaldía de una importante ciudad, se le solicita implementar un algoritmo que permita a los pasajeros encontrar la ruta más corta entre dos estaciones del metro usando dos estrategias diferentes.

Se le suministrará un mapa de la red de metro, y debe determinar la ruta con menos acciones (estaciones de parada) entre dos estaciones usando BFS e IDS.

NOTA: Como un plus adicional en su informe, presentará los resultados de ambos métodos en términos de tiempo de ejecución y memoria utilizada.

Mapa de la Red de Metro

Este metro tiene 10 estaciones, las cuales están conectadas de la siguiente manera:

- Estación A está conectada a Estación B y Estación C.
- Estación B está conectada a Estación A, Estación D, y Estación E.
- Estación C está conectada a Estación A y Estación F.
- Estación D está conectada a Estación B y Estación G.
- Estación E está conectada a Estación B, Estación H y Estación I.
- Estación F está conectada a Estación C y Estación J.
- Estación G está conectada a Estación D.
- Estación H está conectada a Estación E.
- Estación I está conectada a Estación E y Estación J.
- Estación J está conectada a Estación F y Estación I.

Definición del Problema

1. **Estado Inicial:** La estación donde comienza el pasajero.
2. **Estado Objetivo:** La estación a la que el pasajero quiere llegar.
3. **Acciones:** Desde cada estación, el pasajero puede moverse a cualquier estación conectada directamente.
4. **Espacio de Estados:** Todas las posibles combinaciones de estaciones y movimientos entre ellas.
5. **Modelo de Transición:** El estado resultante después de moverse de una estación a otra.

Ejercicio

1. Realice el diseño del grafo considerando un costo de igual valor entre estaciones.
2. **Implementación:** Haz las definiciones pertinentes para la clase Node y Problem así como también la definición de actions.
3. **Algoritmos:** Implementa dos versiones del algoritmo de búsqueda:
 - a. **Breadth-First Search (BFS):** Deberás implementar un algoritmo BFS que explore todas las rutas posibles de manera uniforme, nivel por nivel, para encontrar la ruta más corta.
 - b. **Iterative Deepening Search (IDS):** Deberás implementar un algoritmo IDS que combine la profundidad con la búsqueda en anchura, aumentando progresivamente la profundidad hasta encontrar la solución.
4. **Comparación:** Ejecuta ambos algoritmos para encontrar la ruta más corta entre las estaciones A y J. Compara los resultados obtenidos en términos de tiempo de ejecución y memoria.
5. Explica las diferencias encontradas entre ambos algoritmos.