



Introducción a la Inteligencia Artificial

Proyecto Aplicativo

Profesor:

Ing. Julio Omar Palacio Niño, M.Sc.

palacio_julio@javeriana.edu.co

Temática:

Algoritmos de búsqueda,

Integrantes:

Grupos máximo de 4 personas

Problema

El objetivo del siguiente proyecto es la construcción de un programa que permita encontrar la solución de un laberinto, para lo cual se emplearía las estrategias de búsqueda a ciegas y búsqueda heurística.

Lectura de Laberinto

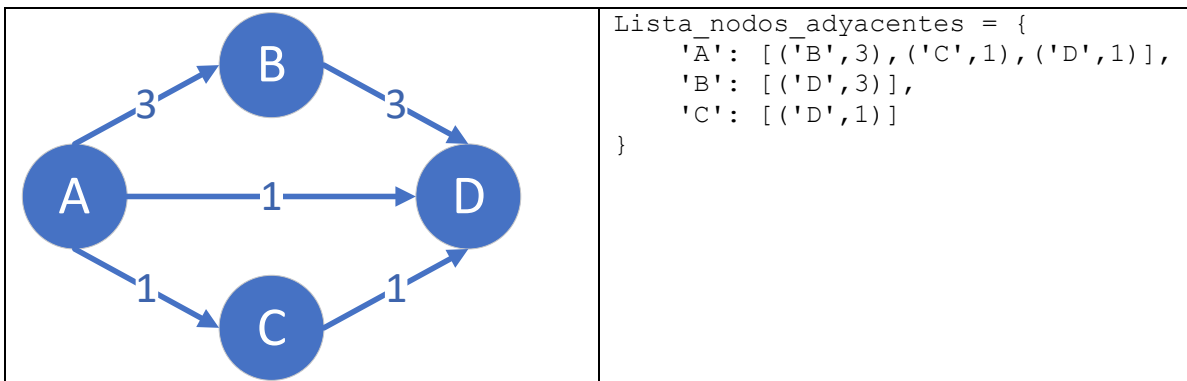
Para la lectura del laberinto ha de realizarse por medio de una matriz en Python de tamaño NxN, los espacios son 0, las paredes 1, salida 2, meta 3

2	1	1	1	0	0	1	1	1	1
0	0	0	1	1	1	0	0	1	0
1	1	0	0	0	1	0	1	1	1
0	1	1	1	0	1	0	1	0	1
1	1	0	1	0	1	0	0	1	1
0	0	0	1	0	3	1	0	1	0
1	1	1	1	1	0	1	0	1	1
1	0	0	0	0	0	1	1	1	0
1	1	1	1	1	1	0	1	0	1
1	0	1	0	1	0	1	1	1	1
1	0	1	0	1	0	1	1	1	1

Laberinto=[
[2, 1, 1, 1, 0, 0, 1, 1, 1, 1]
[0, 0, 0, 1, 1, 1, 0, 0, 1, 0]
[1, 1, 0, 0, 0, 1, 0, 1, 1, 1]
[0, 1, 1, 1, 0, 1, 0, 1, 0, 1]
[1, 1, 0, 1, 0, 1, 0, 0, 1, 1]
[0, 0, 0, 1, 0, 3, 1, 0, 1, 0]
[1, 1, 1, 1, 1, 0, 1, 0, 1, 1]
[1, 0, 0, 0, 0, 1, 1, 1, 0, 1]
[1, 1, 1, 1, 1, 1, 0, 1, 0, 1]
[1, 0, 1, 0, 1, 0, 1, 1, 1, 1]

Lectura de Grafo

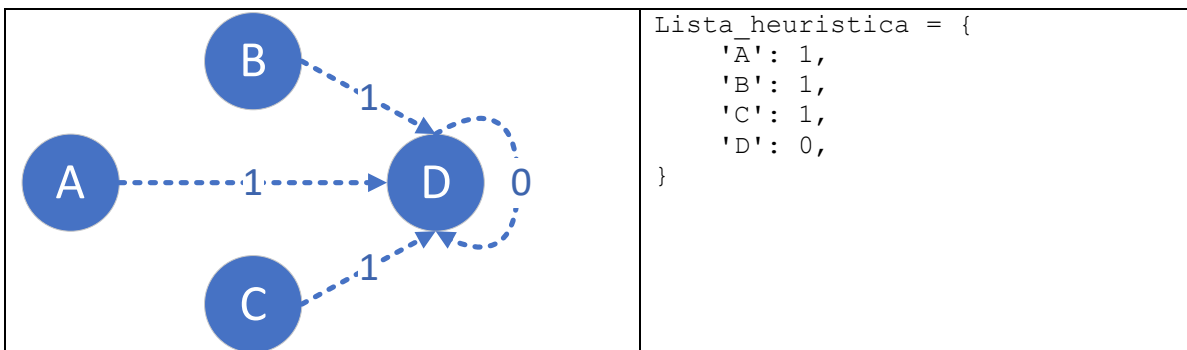
Para manipular el laberinto como uno de los algoritmo se debe realizar una transformación a un grafo, para esto se recomienda usar una lista de adyacencia, a continuación, se presenta un ejemplo grafo ponderado como una lista de adyacencia



Heurística

Recuerde que la función heurística se evaluar del nodo actual de análisis al nodo final, por lo cual una posible solución es la construcción de una tabla con la heurística hacia el nodo objetivo

Heurística: $h(n) = 1$, objetivo D



Estrategia

Para dar solución se debe realizar los siguientes pasos

- Transformar el laberinto a un grafo
- Sobre el grafo aplicar un algoritmo de búsqueda ciega (profundidad, anchura)
- Calcular heurística
- Sobre el grafo + heurística aplicar algoritmo de búsqueda A*

Implementación

El código del proyecto deberá estar totalmente desarrollado en Python para lo cual deberá estar compuesto en los siguientes componentes

- lectura del laberinto (matriz)
- transformación matriz a grafo (nodos, enlaces, pesos)
- lectura de la función heurística
- lectura de los nodos iniciales y finales
- impresión de la ruta calculada

Para la construcción del proyecto se entregará como base un código en Python donde deberá agregar lo necesario para el funcionamiento adecuado de la clase

El programa deberá implementar

- función primero en profundidad
- función primero en anchura
- función de cálculo de heurística
- función A*

como función principal deberá pasar los siguientes insumos

- matriz laberinto
- dimensiones N
- coordenada salida
- coordenada meta

como resultado ha de entregar la ruta que permita dar la solución al laberinto

Entregables

- Documento tipo informe con el desarrollo del problema, como afrontaron cada uno de los desafíos
- Código fuente debidamente comentariado
- Todo se entregará en campus virtual

Sustentación

- Se procederá a realizar una pequeña demostración del funcionamiento del proyecto al resto de la clase en una fecha designada en clase