

UNIVERSIDAD DEL VALLE DE GUATEMALA

Inteligencia Artificial



Diego Jose Valdez Soto 21328

GUATEMALA, 3 de marzo de 2025

Resultados

Laberinto 1

Algoritmo	Nodos Visitados	Tiempo de Ejecución (Segundos)	Largo del camino
BFS	31500	0.075592	250
DFS	15751	0.055242	15750
Greedy (Euclidiana)	251	0.003525	250
Greedy Manhattan	215	0.002852	250
A* (Euclidiana)	15876	0.101501	250
A* (Manhattan)	15876	0.124640	250

Laberinto 2

Algoritmo	Nodos Visitados	Tiempo de Ejecución	Largo del camino
BFS	30391	0.055820	250
DFS	25199	0.057534	5210
Greedy (Euclidiana)	591	0.004557	280
Greedy Manhattan	2273	0.011509	366
A* (Euclidiana)	13559	0.086683	250
A* (Manhattan)	11253	0.071332	250

Laberinto 3

Algoritmo	Nodos Visitados	Tiempo de Ejecución	Largo del camino
BFS	27794	0.056333	124
DFS	7256	0.022688	5872
Greedy (Euclidiana)	168	0.001512	136
Greedy Manhattan	125	0.001004	124
A* (Euclidiana)	2850	0.020835	124
A* (Manhattan)	1054	0.00810	124

Discusión de implementación y de resultados.

¿Qué desafíos de implementación presentó cada algoritmo?

Los desafíos que se encontraron la implementación de cada algoritmo fueron los siguientes. Para el BFS el problema que se encontró durante la implementación fue que como se visitan demasiados nodos por lo que se buscó la manera más eficiente de realizar el código para que este no tarde demasiado y que no consuma demasiados recursos. En el DFS el problema fue la cola LIFO ya que esta tiene que funcionar correctamente para que el algoritmo pueda funcionar de manera eficiente y no se quede en un ciclo infinito ya que durante la implementación ocurría muchas veces que el ciclo no terminaba y se quedaba en un ciclo infinito. En el algoritmo de Greedy y A* los problemas fue que utilizaran las heurísticas correctamente ya que muchas veces no utilizaba bien estas por lo que la implementación fue un poco costosa.

¿En qué casos es mejor cada algoritmo en comparación con el resto?

Depende que categoría o que es lo que se está valorando con más valor, en términos de eficiencia de nodos visitados el mejor algoritmo es el Greedy por su naturaleza de buscar siempre la ruta óptima sin que le importe el costo ya recorrido por esto mismo es que revisa menos nodos. En términos de tiempo de ejecución el mejor es el Greedy por esto mismo ya que recorre menos nodos y tiene que realizar menos cálculos que el A* por lo que su tiempo de ejecución es más rápido ya que el objetivo de este algoritmo es buscar la mejor ruta posible. En términos del camino óptimo el mejor son A* y BFS ya que todos los caminos que encontraron eran el óptimo y con de menor longitud. Lo que hace que el A* sea mejor que el BFS es la cantidad de nodos que visitan cada uno mientras que el A* visita una cantidad baja de nodos el BFS revisa una gran cantidad de estos haciendo que no sea tan eficiente en comparación con el algoritmo A*. En términos generales los dos mejores algoritmos son A* y Greedy siendo estos dos de búsqueda informada, pero si hay que escoger uno sería el A* ya que este prioriza el mejor camino y no la velocidad como el algoritmo de Greedy ya que hay casos como en el laberinto 2 donde encontró la solución en menos nodos y menos tiempo, pero teniendo un largo del camino superior al algoritmo de A*

¿Cómo interpreta los resultados de las métricas obtenidas?

Se interpretan de que dependiendo la estadística que se quiere valorar hay ciertos algoritmos que son mejores que otros, como se mencionó el objetivo principal es encontrar una solución rápida y que no consuma tantos recursos por lo que se hizo la valoración de que el algoritmo A* sea el mejor algoritmo de búsqueda para estos laberintos siendo este el más rápido teniendo en cuenta la cantidad de nodos y que fue el que encontró siempre el mejor camino siempre.

¿Encontró soluciones óptimas?

Si, se encontraron rutas óptimas entre el algoritmo de BFS y A* se encontraron las rutas óptimas en los 3 laberintos por lo que estos algoritmos a la hora de buscar soluciones a los laberintos de forma óptima son los mejores ya que estos siempre encuentran el mejor camino a la salida incluyendo siempre los puntos aleatorios generados.