

Inteligencia Artificial 2026

Pegundo Proyecto

08.abril.2026

Segundo Proyecto: Algoritmos de Búsqueda

En este proyecto haremos una comparación del desempeño de diferentes algoritmos de búsqueda para encontrar caminos a través de laberintos. La idea es aplicar un conjunto de técnicas de IA para resolver diferentes problemas.

1 Generación aleatoria de laberintos

En un grid de tamaño $M \times N$ (M filas, N columnas), desarrollar una función que genere laberintos de forma aleatoria. Para ello, deberá implementar al menos 2 de los algoritmos para encontrar el árbol de expansión mínima comentados en clase:

- algoritmo de Kruskal,
- algoritmo de Prim,
- algoritmo de Boruvka,
- algoritmo de División Recursiva.

Generar visualizaciones de la construcción de estos laberintos aleatorios, comparando la construcción de ambos algoritmos implementados. Por ejemplo, puede utilizar de referencia las siguientes visualizaciones:

- <https://www.youtube.com/watch?v=h8VWRDJBf1k>,
- <https://www.youtube.com/watch?v=Kyoep91w7NE>,
- <https://www.youtube.com/watch?v=VBNZJD4wCnE>,
- <https://www.youtube.com/watch?v=vmWSnkBVvQ0>.

2 Solución de un laberinto aleatorio

Dado un laberinto en un grid de tamaño 60×80 (generado por los algoritmos del Problema 1), desarrollar una función que resuelva la ruta más corta entre la entrada del laberinto: celda $A = (1, 1)$ y la salida: celda $B = (60, 80)$.

Mostrar en la visualización, las celdas exploradas por su algoritmo, así como el camino encontrado. Al final de la simulación, muestre una estadística de la longitud del camino encontrado, y de la cantidad de nodos explorados en la búsqueda. Puede elegir implementar de entre los algoritmos: BFS, DFS, Cost Uniform Search, A^* .



3 Comparación de algoritmos de búsqueda

En un grid de tamaño 45×55 , generar diferentes escenarios con laberintos aleatorios, y comparar el desempeño de los siguientes algoritmos de búsqueda en la solución de estos laberintos:

- BFS,
- DFS,
- Cost Uniform Search (Dijkstra),
- A^* .

Para ello, deberá generar $K = 25$ laberintos aleatorios distintos, y en cada uno de ellos deberá elegir una posición inicial A y una posición final B arbitrarias, (que estén al menos a una distancia Manhattan de 10 unidades).

Para cada una de las K simulaciones, hacer lo siguiente: Elaborar visualizaciones de la solución de una tabla con el **conteo de nodos explorados** por cada algoritmo, el **tiempo de ejecución**, así como de la **longitud de la ruta óptima obtenida** por cada uno. Ordenar los algoritmos del 1 al 4 para determinar en cada escenario cuál fue el mejor.

Finalmente, en su informe elaborar una tabla resumen que indique el ranking o desempeño promedio de los 4 algoritmos en todos los escenarios aleatorios. Puede tomar de referencia la siguiente comparación:

- <https://www.youtube.com/watch?v=GC-nBgi9r0U> (ver a partir del minuto 3:44).



Se espera observar varias simulaciones o escenarios probados (pueden añadirle obstáculos, pesos distintos a regiones del mapa, etc.)

Algunos recursos importantes que pueden servir para inspirar sus visualizaciones:

- <https://clementmihailescu.github.io/Pathfinding-Visualizer/>
- <https://qiao.github.io/PathFinding.js/visual/>

Entregables

1. Reporte técnico del proyecto
2. Código y recursos utilizados.

El código debe subirse a Canvas. Por favor, todo el código debe ser empaquetado en un archivo .zip o .rar, y entregarlo junto con el reporte en formato .pdf.

Tomar en cuenta que en el reporte no debe ir código. El reporte debe entregarse fuera del archivo comprimido, para facilitar la lectura. **El no cumplimiento de lo anterior tendrá una penalización sobre su nota del proyecto.**

El reporte debe incluir:

- Explicación breve de los algoritmos usados para generar los laberintos.
- Explicación breve de los algoritmos para encontrar la ruta más corta de salida (resolver el laberinto).
- Descripción técnica de la implementación de los algoritmos: arquitectura o clases utilizadas para implementar el árbol de búsqueda, implementación de los métodos para cada algoritmo, Detalles técnicos relevantes en la implementación.
- Visualizaciones y animaciones de los resultados en la generación de laberintos (Problema 1).
- Visualizaciones y animaciones de los caminos y región explorada por los algoritmos de búsqueda de ruta más corta (Problema 2).
- Visualizaciones y tablas comparativas para las soluciones del Problema. Tome en cuenta que se espera ver diferentes escenarios testados.
- Conclusiones según los resultados de la comparación.

Fecha de Entrega: lunes 3 de mayo.

Se asignará una hora para que cada grupo presente sus resultados.