Comparative Grid‑Map Path Planning

Colab Link:
https://colab.research.google.com/drive/1uwWu1dMoDss4MRODtKDdhUKVm2UJkIpN?usp=sharing

Kaggle Maze Dataset:
https://www.kaggle.com/datasets/mexwell/maze-dataset
(used the code for imperfect maze from there)

1. Description of Implementation Choices

This project compares two path planning algorithms — Dijkstra's and A* — on three different types of 2D occupancy grid maps: sparse map, maze map, and cost field map. The algorithms were implemented using a priority queue to manage the frontier of nodes being explored. Each map type has distinct characteristics:
- Sparse Map: Large open areas with few obstacles.
- Maze Map: Dense maze-like structure with narrow passages.
- Cost Field Map: Variable traversal costs across the grid.

The goal was to evaluate each algorithm's performance in terms of computation time, path length, and reliability (success rate).

2. Benchmark Results

Benchmark results based on 5 trials per map:

Algorithm	Map Type	Avg Time (ms)	Avg Path Length	Success Rate
Dijkstra	Sparse	2.52	49.0	5/5
Dijkstra	Maze	5.61	107.0	5/5
Dijkstra	Cost Field	3.45	49.0	5/5
A*	Sparse	2.60	49.0	5/5
A*	Maze	1.86	107.0	5/5
A*	Cost Field	3.00	49.0	5/5

The average time taken for Dijkstra is 3.86s and for A* is 2.49s in finding the shortest path.

3. Analysis and Comparison

The benchmark results indicate that A* is generally faster than Dijkstra's algorithm on average, particularly in structured environments like maze maps. While both algorithms yield the same path length on sparse and cost field maps, A* achieves this more efficiently due to its heuristic guidance, which reduces unnecessary node exploration.

Specifically, A* outperforms Dijkstra on the maze map (1.86 ms vs. 5.61 ms) while producing identical path lengths. On sparse and cost field maps, the performance is similar, but A* remains marginally quicker.

I have also showed the explored graph for both Dijkstra and A* algorithm. It shows that, Dijkstra explores more paths than A* algorithm, and that is one of the main reason why A* is faster than Dijkstra.

In terms of reliability, both algorithms consistently find the shortest path in all scenarios without failure (5/5 success rate). No algorithm failed due to timeout or impassable maps under the current test conditions.

In summary:
- A* is the fastest on average.
- Both consistently find the shortest path.
- Neither algorithm failed in any map condition.