# Robot Maze Pathfinding Solver-Project proposal

## 1 Introduction

This project focuses on implementing and comparing different search strategies to guide a robot through a grid-based maze from a start point to a goal point while avoiding obstacles.

The project aims to demonstrate how uninformed and informed search algorithms differ in terms of efficiency, optimality, and performance.

## 2 Problem Statement

The problem is to find a valid and efficient path for a robot to move from a predefined start cell to a target cell in a 2D maze environment.

The maze consists of:

- Free cells where the robot can move.
- Obstacles that block movement.

The robot can move in four directions:

- Up
- Down
- Left
- Right

The challenge is to compare multiple AI search algorithms and analyze their performance based on:

- Number of nodes explored
- Execution time
- Path optimality

## 3 Objectives

The main objectives of this project are:

- To implement different AI search algorithms.
- To visualize the pathfinding process.
- To compare the performance of uninformed and informed search techniques.
- To analyze the advantages and limitations of each algorithm.

## 4 Algorithms Used

### ◇ 4.1 Breadth-First Search (BFS)

In our maze, BFS explores the grid level by level starting from the start position. The algorithm checks all neighboring cells equally before moving deeper into the maze.

**Result in our problem:**

- BFS successfully found the **shortest path** to the goal.
- It explored a **large number of nodes** because it expands all possible paths at the same depth.
- The execution time was higher compared to informed search algorithms.

**Conclusion:**
BFS guarantees an optimal solution, but it is inefficient in terms of time and memory for larger mazes.

### ◇ 4.2 Depth-First Search (DFS)

DFS explores one path deeply before backtracking to explore alternative paths.
In our maze, DFS followed a single direction until it reached a dead-end or the goal.

**Result in our problem:**

- DFS reached the goal faster in some runs.

- The path found was **not always the shortest**.
- It explored fewer nodes than BFS but produced a suboptimal path in some cases.

**Conclusion:**
DFS is memory-efficient but unreliable when path optimality is required.

## ◇ 4.3 A* Search Algorithm

A* combines path cost and heuristic information to guide the search process.
In this project, the **Manhattan Distance heuristic** was used, which is suitable for grid-based environments.

**Result in our problem:**

- A* explored significantly **fewer nodes** than BFS and DFS.
- It consistently found the **optimal (shortest) path**.
- The execution time was lower compared to BFS.

**Conclusion:**
A* achieved the best balance between efficiency and optimality, making it the most suitable algorithm for this maze problem.

## ◇ 4.4 Greedy Best-First Search (GBFS)

GBFS selects the next node based only on the heuristic value, without considering the total path cost.

**Result in our problem:**

- GBFS reached the goal **very quickly**.
- It explored the **least number of nodes**.
- However, the path found was **not guaranteed to be optimal**.

**Conclusion:**
GBFS is fast but unreliable when the shortest path is required.

## 5 Comparative Analysis

| Algorithm | Optimal Path | Nodes Explored | Execution Time |
|---|---|---|---|
| **BFS** | Yes | High | Medium |
| **DFS** | No | Medium | Fast |
| **A*** | Yes | Low | Fast |
| **GBFS** | No | Low | Very Fast |

## 6 Final Discussion

The comparison shows that uninformed search algorithms explore the maze without guidance, resulting in higher node expansion.
In contrast, informed search algorithms use heuristics to reduce unnecessary exploration.

Among all algorithms, **A*** proved to be the most effective solution for our robot maze problem due to its ability to find the optimal path with fewer explored nodes.