

# Project Proposal

## Solving Maze Pathfinding Problem with Search Algorithms

### PROJECT DESCRIPTION

The maze problem is a classic pathfinding challenge in artificial intelligence.

The maze is made of a grid that contains open cells the agent can move through, and walls that the agent cannot cross.

The goal is to move from a start point to a goal point without entering any blocked cells.

This problem is useful because it shows the differences between uninformed and informed search methods in a clear way.

### WHY THIS PROBLEM

The maze problem was chosen because it makes it easy to compare how different search algorithms work. All algorithms face the same maze, so their behavior can be seen and compared fairly. It supports comparison between uninformed and informed search techniques.

### IMPLEMENTATION APPROACH

#### State Representation:

The state is represented as a coordinate pair  $(x, y)$  that shows the agent's current position in the maze.

#### Successor Function:

The successor function produces all valid neighboring cells the agent can move to. These include the four main directions: up, down, left, and right, as long as the cell is inside the maze and not a blocked cell.

#### Cost Function:

Each move from one cell to the next has a cost of 1. If the maze contains different types of terrain, each type can have a different cost.

#### Heuristic Functions (for A\* Search):

These heuristics estimate how far the current position is from the goal:

- Manhattan Distance:  $|x_1 - x_2| + |y_1 - y_2|$
- Euclidean Distance: The straight-line distance to the goal
- Diagonal Distance: Useful when diagonal movement is possible (or for estimation)

***Among these heuristics, Manhattan Distance is used in this project because the maze allows movement only in four directions (up, down, left, and right).***

### USED ALGORITHMS

#### Informed Search Algorithms:

- Breadth-First Search (BFS): guaranteed to find the shortest path
- Depth-First Search (DFS): not guaranteed to find the shortest path but helps illustrate uninformed exploration behavior

#### Uninformed Search Algorithms:

- A\* Search: guide the search toward the goal

## EXPECTED RESULTS

For a 20×20 maze with multiple obstacles, the algorithms are expected to show the following general behavior:

Algorithm	Path Length	Nodes Explored	Time
BFS	40~	230~	0.001 ~
DFS	100~	220~	0.0006~
A*	40~	55~	0.0002~

## DELIVERABLES

1. Maze generator.
2. Visualization of maze with solution path.
3. Implementation of all algorithms.
4. Performance comparison.
5. Analysis of heuristic effectiveness

## TEAM MEMBERS

- Jana Ahmed Ghareb -> Maze generation
- Jana Hosny Dawood -> A\* search algorithm implementation
- Kenzy Hossam Hewedy -> BFS implementation
- Abdelraouf Mahmoud Abdelraouf -> DFS implementation
- Khaled Waleed Ramadan -> Algorithm comparison