**Name:** Aqsa Gillani  
**Roll No:** 064  
**Course:** Artificial Intelligence (AI) Lab  
**Semester:** Fall 2025

*1. A Algorithm\**

**Objective:**  
To implement the A\* (A-star) search algorithm using a tree structure to find the shortest path in a grid with obstacles.

**Description:**  
A\* is a **best-first search algorithm** that finds the shortest path from a start position to a goal position.

* Each position is represented as a **Node** with parent and children pointers.
* The algorithm uses a **cost function f = g + h**:
  + g → actual cost from start node to current node.
  + h → estimated cost from current node to goal (heuristic, e.g., Manhattan distance).
* Nodes are expanded like a **tree**: each node keeps a list of children representing possible moves.
* The algorithm avoids obstacles and ensures the shortest path is found efficiently.

**Code Explanation:**

* A **Node class** is defined with attributes: position, parent, children, g, h, and f.
* The **heuristic function** calculates the Manhattan distance to the goal.
* Two lists are used:
  + **Open list**: Nodes to be explored (frontier).
  + **Closed list**: Nodes already explored.
* Steps of the algorithm:
  + Start from the root node (start position) and add it to the open list.
  + Select the node with the **lowest f value** from the open list.
  + Move it to the closed list and expand its **children** (possible moves).
  + For each child:
    - Check grid boundaries and obstacles.
    - Calculate g, h, f.
    - Skip if already explored or if a better path exists in the open list.
  + Add valid children to the **tree** and open list.
  + Repeat until the **goal node** is reached.
  + Reconstruct the **shortest path** by following parent pointers from goal to start.

**Output:**

