**A\* ALGORITHM:**

**Node Class**:

* Represents a point on the grid.
* Stores g (cost from start), h (heuristic to goal), and f = g + h.
* Also keeps track of its parent for path reconstruction.

**Heuristic Function**:

* Uses **Manhattan distance** to estimate the cost from a node to the goal.

**Find Lowest f-score Function**:

* Manually searches the open\_list for the node with the lowest f value.

*A Algorithm Execution*\*:

* **Initialization**: Creates start\_node and end\_node.
* **Loop**:
  + Finds the node with the lowest f in open\_list.
  + Moves it to closed\_list.
  + If it’s the goal, **backtracks** to reconstruct the path.
  + Expands valid neighbors (not obstacles or out-of-bounds).
  + Updates g, h, f values and adds neighbors to open\_list.
* **Stops** when reaching the goal or if no path is found.

**Example Usage**:

* Given a **5×5 grid** with obstacles (1), finds the shortest path from (0,0) to (4,4).