

**Name: Mahlika**

**Roll: Su92BSAIM059**

**Subject: AI LAB**

**Task :7**

**A\* Search Algorithm Documentation**

**Introduction**

This program uses the A\* (A-Star) search algorithm to find the shortest path between two nodes in a graph. It combines the real cost of moving from node to node with a guessed cost to reach the goal. A priority queue helps the program pick the best node to explore first, which makes the search faster and more efficient.

**Graph Structure**

The graph is saved as an adjacency list. Each node has a list of neighbors and the cost to move to them. The heuristic dictionary stores estimated distances from each node to the goal, helping the algorithm choose better paths.

**A\* Function Explanation**

The a\_star() method starts with the open list, cost dictionary g, parent dictionary, and a closed set. It removes the node with the lowest total cost (real cost + heuristic) from the priority queue. Then, it checks all neighboring nodes and updates their cost if a shorter path is found. When the end node is reached, the loop stops, and the path reconstruction function is used.

# ****Enhancement in This Code****

A major enhancement in this implementation is the use of the heapq priority queue. Instead of checking all available nodes manually to find the one with the lowest cost, the priority queue automatically selects the best option efficiently. This greatly reduces time complexity and improves performance, especially for larger graphs. Additionally, the algorithm prevents revisiting nodes that were already processed, which saves memory and processing time. Overall, this enhancement makes the A\* algorithm faster, smarter, and more efficient.

**Key Features and Improvements**

The use of heapq makes choosing the lowest-cost node fast. Cost checks prevent updating worse paths. The closed set keeps the program from visiting the same node again, making the search faster and more accurate.

**Conclusion**

Overall, this A\* implementation is simple, fast, and reliable. By using cost and heuristic values together, it can quickly find the best route in many real-world pathfinding situations.