**Assignment No: 4**

**Problem Statement:-**

Implement the A\* Algorithm for an application.

**Theory:-**

The *A Algorithm*\* is a widely used pathfinding and graph traversal algorithm. It is optimal and complete, meaning it always finds the shortest path between the starting node and the goal node, provided an admissible heuristic is used.

* *A Algorithm*\*: This combines **Dijkstra's Algorithm** (which finds the shortest path) with a **Greedy Best-First Search** (which uses heuristics to guide the search).
* **f(n) = g(n) + h(n)**:
  + g(n) is the cost from the start node to node n.
  + h(n) is the estimated cost from node n to the goal (the heuristic function).
  + f(n) is the total estimated cost of the cheapest solution through node n.

**Methodology:-**

1. **Problem Representation**:
   * The problem is modeled as a graph with nodes representing states and edges representing transitions between states. For example, a grid-based map can represent locations where A\* finds the shortest route from a start point to a destination.
2. **Algorithm Steps**:
   * **Initialize** the open list (nodes to be explored) and the closed list (nodes already explored).
   * **Start** from the initial node and calculate f(n) for all neighboring nodes.
   * **Expand** the node with the lowest f(n) value, moving closer to the goal.
   * Continue expanding nodes, updating their costs until the goal is reached.
3. **Heuristics**:
   * For example, in a grid-based system, **Manhattan distance** or **Euclidean distance** can be used as heuristics to estimate h(n).
4. **Application Examples**:
   * **Game AI**: Finding the shortest path for a game character.
   * **Robot Navigation**: Pathfinding for robots in a grid-like environment.
   * **GPS Systems**: Shortest route finding in maps for navigation systems.

**Conclusion:-**

We implemented the A\* algorithm for a pathfinding application, demonstrating its efficiency in finding the shortest path using a combination of cost functions and heuristics.