Modern Education Society's
Wadia College of Engineering, Pune-01
Department of Computer Engineering

NAME OF STUDENT:	**CLASS:**
SEMESTER/YEAR:	**ROLL NO:**
DATE OF PERFORMANCE:	**DATE OF SUBMISSION:**
EXAMINED BY:	**EXPERIMENT NO: 02**

TITLE: Implement A Star Algorithm For 8 Puzzle Problem

PROBLEM STATEMENT: To solve 8-puzzle problem using A* algorithm. Assume any initial configuration and define goal configuration clearly

OBJECTIVES:

1. To understand, what is state space search and its importance.
2. To understand the implementation of A* algorithm

THEORY:

- A* Algorithm is one of the best and most popular techniques used for path finding and graph traversals.
- A lot of games and web-based maps use this algorithm for finding the shortest path efficiently.
- It is essentially the best first search algorithm.

A* Algorithm works as-

- It maintains a tree of paths originating at the start node.
- It extends those paths one edge at a time.
- It continues until its termination criterion is satisfied.

A* Algorithm extends the path that minimizes the following function-

$$f(n) = g(n) + h(n)$$

Here,

- 'n' is the last node on the path
- g(n) is the cost of the path from start node to node 'n'
- h(n) is a heuristic function that estimates cost of the cheapest path from node 'n' to the goal node

Algorithm-

- The implementation of A* Algorithm involves maintaining two lists- OPEN and CLOSED.
- OPEN contains those nodes that have been evaluated by the heuristic function but have not been expanded into successors yet.
- CLOSED contains those nodes that have already been visited.

The algorithm is as follows-

Step-01:

- Define a list OPEN.
- Initially, OPEN consists solely of a single node, the start node S.

Step-02:

If the list is empty, return failure and exit.

Step-03:

Remove node n with the smallest value of f(n) from OPEN and move it to list CLOSED.

- If node n is a goal state, return success and exit.

Step-04:

Expand node n.

Step-05:

If any successor to n is the goal node, return success and the solution by tracing the path from goal node to S.

- Otherwise, go to Step-06.

Step-06:

For each successor node,

- Apply the evaluation function f to the node.
- If the node has not been in either list, add it to OPEN.

Step-07:

Go back to Step-02.

Points to remember:

- A* algorithm returns the path which occurred first, and it does not search for all remaining paths.
- The efficiency of A* algorithm depends on the quality of heuristic.
- A* algorithm expands all nodes which satisfy the condition f(n).

Advantages:

- A* search algorithm is the best algorithm than other search algorithms.
- A* search algorithm is optimal and complete.
- This algorithm can solve very complex problems.

Disadvantages:

- It does not always produce the shortest path as it is mostly based on heuristics and approximation.
- A* search algorithm has some complexity issues.
- The main drawback of A* is memory requirement as it keeps all generated nodes in the memory, so it is not practical for various large-scale problems.

Applications of A* Algorithm

The A* algorithm is widely used in various domains for pathfinding and optimization problems. It has applications in robotics, video games, route planning, logistics, and artificial intelligence. In robotics, A* helps robots navigate obstacles and find optimal paths.

Questions:

1] Find the most cost-effective path to reach the final state from initial state using A* Algorithm. Given an initial state of a 8-puzzle problem and final state to be reached-

Initial State

Final State

2] Explain Admissibility and Consistency of Heuristic Function in A* Algorithm.