

AIR ASSIGNMENT 1

PROBLEM STATEMENT:

Solve 8 puzzle problem using A* algorithm. Assume any initial configuration & define goal configuration clearly.

OBJECTIVES:

- 1] Understand A* algorithm.
- 2] Understand searching algorithms for 8 puzzle problem.

OUTCOMES:

Solve 8 puzzle problem using A* algorithm.

SW & HW REQUIREMENTS:

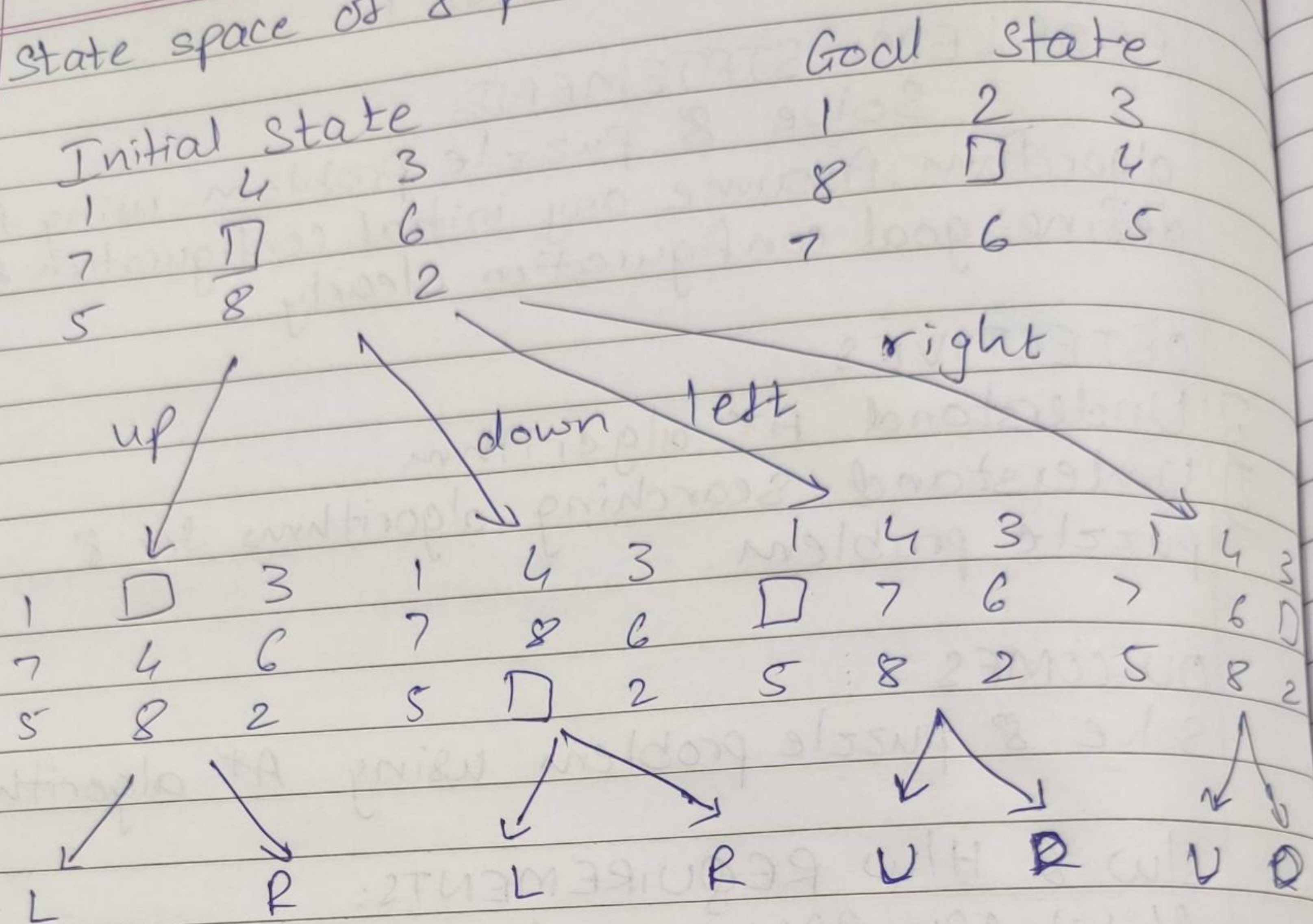
64 bit CPU, RAM, UNIX/LINUX based OS, JVM, IDE, keyboard, mouse, monitor.

THEORY:

8 puzzle problem.

- 1] The puzzle is divided into $\sqrt{N+1}$ rows & $\sqrt{N+1}$ columns.
- 2] So it will have 3 rows & 3 columns, 8 tiles & one empty space where the tile can be moved.
- 3] Start the goal configurations (also called state) as the puzzle can be solved by moving the tiles one by one in single empty space & thus achieving the goal configuration.

State space of 8 puzzle problem:



A* Algorithm:

- i) The A* algorithm integrates characteristics of uniform cost search & heuristic based search to find the optionally efficient solution.
- ii) The key feature of A* is that it keeps track of each visited node which helps in ignoring the already visited nodes, as well as a list of nodes yet to be explored.
- iii) So we use the two lists normally open list & closed list.
- iv) Open list contains all the nodes that are being generated and are not existing in closed list.

The metric used to determine optimality of a node is the f-score.

$$f\text{-score} = \underbrace{h\text{-score}} + g\text{-score}$$

how far
goal node is



no. of nodes traversed
from start node to
current node.

The h-score is the Manhattan distance (the distance b/w two points measured along axes at right angles).

$$h\text{-score} = |x_1 - x_2| + |y_1 - y_2|$$

CONCLUSION:

8 puzzle problem was successfully solved & A* algorithm was implemented to solve it.