LP I

Artificial Intelligence and Robotics
Assignment B1

Date of Completion :- 9.10.2020

Title:- Puzzle.

Problem Statement:- Solve 8-puzzle problem using A* algorithm. Assume any initial configuration and define goal configuration clearly.

OR

• Solve following 6-tile problem stepwise using A* algorithm,

Initial Configuration

B	W	B	W	B	W

Final Configuration

B	B	B	W	W	W

Constraints: Tiles can be shifted left or right 1 or 2 positions with cost 1 and 2 respectively.

Objectives :- Understand 8 puzzle problem
Understand A* algorithm

Outcome :- Students will be able to implement 8 puzzle problem using A* algorithm.

Requirements:- Ubuntu OS, python.

Theory:-

• It is a heuristic search algorithm for finding paths.

1) Consider a square ~~grid~~ grid having many obstacles and we are given a starting cell and target cell.
2) We want to search target cell from the starting cell as quickly as possible.
3) At ~~Start~~ each step, A* algorithm picks the node according to a value 'f' which is equal to sum of 'g' and 'h'.
4) At each step it picks the node cell having least 'f' and process that node.

$$f = g + h.$$

g → movement cost to move from the starting point to a given grid following the path generated to get there.
h → movement cost (estimated) to move from that given grid square on the grid to the final destination. This is often referred to as the heuristic which is ~~too~~ nothing but a kind of smart guess.

Algorithm

```
1. Initialize the open list.
2. Initialize the closed list.
3. put the starting node on the open list
4.  while the open list is not empty
    1) find the node with the leaf $f$ on the open list. Call it 'q'
    2) pop 'q' off the open list.
    3) generate 'q''s successors.
    4) for each successor
            a) if successor is the goal, stop search successor.
               $g = q.g + distance(successor.q)$
               successor.h = distance from goal to successor
               successor.f = successor.g + successor.h
```

```
b) if a node with the same position as successor
   is in the open list which has a lower 'f'
   than successor, skip the successor.
c) if a node with the same position as
   successor is in the CLOSED list which has
   a lower 'f' than successor, skip this
   successor otherwise and the node to the
   open list.
5) end for
6) ~~end~~ push q on the closed list.
4 end while
```

Test Case -

$$\begin{matrix}1 & 2 & x\\ 4 & 5 & 3\\ 7 & 8 & 6\end{matrix} \qquad \qquad \begin{matrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & x\end{matrix}$$

initial final

solved in 18 moves.

$$\begin{matrix}1 & 2 & 3\\ x & 4 & 6\\ 7 & 5 & 8\end{matrix} \qquad \qquad \begin{matrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & x\end{matrix}$$

initial Final

$$\begin{matrix}1 & 2 & 3\\ x & 4 & 6\\ 7 & 5 & 8\end{matrix} \rightarrow \begin{matrix}1 & 2 & 3\\ 4 & x & 6\\ 7 & 5 & 8\end{matrix} \longrightarrow \begin{matrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & x & 8\end{matrix} \rightarrow \begin{matrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & x\end{matrix}$$

Conclusion :- Thus I understood and implement the 8 puzzle problem ~~and~~ using A* algorithm.