

Assignment AJR

Title :-

A* search Algorithm.

Problem Statement.

Solve 8 puzzle problem using A* algorithm.

Objective :-

To learn and implement A* algorithm

Outcome :

Students will be able to implement A* algorithm for 8 puzzle problem.

SW and HW packages.

Operating System : windows or linux
python interpreter

Theory :-

A* algorithm is computer Algorithm that is widely used in path finding and graph traversal the process of plotting an effectively efficient traversable path between multiple points called nodes. It is used for its performance and accuracy.

The key feature of A* is it keeps a track of each visited node which helps in ignoring the nodes that are already visited.

Teacher Signature

It also has a list that holds all nodes that are ~~most~~ are left to be explored and it chooses most optimal node from the list. Thus saving time not exploring unnecessary or less optimal nodes.

So we use two lists open list and close list. The open list contains all nodes that are being generated and are not existing in closed list and each node explored after each neighbour node explore and neighbours in open list is how nodes are expanded.

Each nodes have a pointer to its parent so that at any given point it can retrace the path to the parent. Initially open list holds start node. The next node chosen from list is based on its f score. The node with least f score is picked up and explored.

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

A* uses combination of heuristic value (h score is how far the goal node is) as well as (g-score the number of nodes traversed from start node to current node).

Test case :-

2	8	3
1	6	4
7	-	5

Initial state

2	8	3	2	8	3	2	8	3
1	-	4	1	6	4	1	6	4
7	6	5	-	7	5	7	8	-

2	8	3	2	8	3	2	-	3
-	1	4	1	4	-	1	8	4
7	6	5	7	6	5	7	6	5

-	8	3	2	8	3	-	2	3	2	3	-
2	1	4	7	1	4	1	8	4	1	8	4
7	6	5	-	6	5	4	6	5	7	6	5

1	2	3
-	8	4
7	6	5

1	2	3
8	-	4
7	6	5

Goal state.

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

so $g(n)$ will give level.

so $h(n)$ will give no of nodes

different in start and goal matrix

Complexity :-

The time complexity of A* depends on heuristic.

In the worst case of an unbounded search space, the no of nodes expanded is exponential in depth of solution tree shortest path $O(b^d)$.

Where b is branching function factor.

Conclusion :-

Thus, we have successfully implemented 8 puzzle problem and understood the concept of A* Algorithm.