AIR - Assignment - 3 [LP-I]

Aim:- Implement Best first search & A* algorithm.

Problem Definition:-

Use heuristic search techniques to implement BFS (Best solution but not always optimal) & A* algorithm (always gives optimal solution).

Objectives:-

- To understand heuristic search techniques.
- To understand BFS.
- To understand A* algorithm.

Theory:-

Heuristic search technique:-

A Heuristic search technique is a type of search performed by AI that looks to find a good solution, not necessarily a perfect one out of available options.

This technique makes decisions by ranking every available choice at each branch of a search & then chooses the best option of those presented. Rather than focusing on finding an optimal solution like other search

methods heuristic searching is designed to be quick & therefore finds the most acceptable option within a reasonable time limit or within the allocated memory space.

Algorithm for BFS :-

(1) Place the starting node into the open list.
(2) If the open list is empty, stop & return the "failure".
(3) Remove the node 'n' from 'OPEN' list which has the lowest value of h(n) & places it in the 'CLOSED' list.
(4) Expand the node 'n' & generate the successor of node 'n'.
(5) Check each successor of node 'n' & find whether any node is a goal node or not. If any successor node is goal node then return "Success" & terminate the search, else proceed to step (6).
(6) For each successor node algorithm checks for evaluation function f(n) & then check if node has been either in OPEN or CLOSED list. If the node has not been in list, then add it to "OPEN" list.
(7) Return to step (2).

A* Algorithm :-

A* search is the most commonly known form of BFS, it uses heuristic function h(n) & cost to reach the node n from start state g(n). It has combined features of uniform Cost search & greedy BFS by which it solves problem efficient. A* finds shortest path through search space using h(n).

Conclusion :-

From this assignment, we learnt about heuristic search technique, BFS and A* algorithm.

Reference :-

- Geeksforgeeks.com