* Experiment - 6 (Postlab) :-

Q.1) What is the difference between A* and AO* algorithm?

Ans

A* algorithm	AO* Algorithm
i) It is a best-first search algorithm used primarily for pathfinding and graph traversal. It operates on a graph of nodes (states) connected by edges (actions) with associated costs.	i) AO* (And/Or) is a heuristic search algorithm used for solving graphs that contain both AND nodes and OR nodes, representing different types of decision problems.
ii) Uses a heuristic function 'h(n)' that estimates the cost to reach the goal from node 'n', combined with 'g(n)', the cost from the start node to 'n', guiding the search towards the least costly path.	ii) AO* also uses a heuristic function but is designed to handle graphs where some nodes ("OR nodes") represent choices between alternatives, and other nodes ("AND nodes") require all child nodes to be solved (i.e. multiple subgoals must be achieved).

iii) A* is complete and optimal as long as the heuristic function is admissible (i.e, it never overestimates the actual cost to get to the nearest goal). It expands the node with the lowest 'f' value first and terminates when the goal state is dequeued from the priority queue.	iii) AO* is used for goal directed searches in these complex graphs, where the objective is to find a set of actions (a plan) that achieves a goal. It systematically explores and expands these And/Or graphs by creating a solution graph and optimises it by pruning paths that are not promising.
iv) Typically used in scenarios where a single solution path needs to be found in a space with multiple nodes, such as navigation or route optimization.	iv) More suited for applications in planning and problem solving where decisions are dependent on multiple sub-tasks being completed.

(Q2) Why AO* algorithm only works when heuristic values are underestimated?

Ans:- The AO* algorithm's performance and correctness are heavily dependent on the heuristic values being admissible, meaning they must not overestimate the cost to reach the goal. The reasons are as follows:-

a) Optimality and Completeness:- For AO* to guarantee that the solution found is optimal, the heuristic estimates used to guide the search must not exceed the actual minimal cost required to achieve the subgoals from any node in the graph. Overestimations could lead the algorithm to prune parts of the graph that contains the optimal solution, hence losing both optimality and completeness.

b) Search Efficiency:- Underestimating the cost tends to lead the search optimistically towards the goal, expanding fewer nodes than overestimating, which might lead the search astray or cause premature cutoffs in parts of the graph that might contain viable solutions.

c) Heuristic Driven:- Like A*, AO* prioritizes nodes based on the heuristic value, using it to build and continually refine the solution graph. An accurate or underestimated heuristic helps the algorithm in maintaining a directed and efficient exploration of the And/Or graph.