NAME:- KRUNAL RANK

ADM. NO:- U18CO081

BTECH 3RD YEAR

U18CO081

23/9/20

AIML
Tutorial 6

A* Algorithm	AO* Algorithm
• It is used in pathfinding and graph traversal, also in the process of plotting an efficiently directed path between a number of points called nodes.	• AO* follow a similar procedure but there are constraints travelling specific paths.
• A* traverse the tree in depth and keep moving and adding up the total cost of reaching the cost from current state to goal state and add it to the cost of reaching the current state.	• When those paths are traversed, cost of all paths which originate from the preceding node are added till that level where you find the goal state regardless of the fact whether they take you to the goal state or not.

A star algorithm is heavily used as a replacement for Dijskhra's greedy approach for shortest path. Because of the cumulative heuristic function that includes path distance and heuristic value, A star algorithm performs much better. It is also used in solving state based problems such as Sudoku, 8 puzzle problem, etc.-

AO* algorithm is used for problem reduction into simple binary operations of AND and OR. It is heavily used in decision making systems recommendation systems and decision making systems

Ans 2: A star algorithm works really well ~~in~~ in terms of path finding. But it has memory issues.

~~A~~ Iterative Deepening A* algorithm solves this problem using the DFS approach.

A* star implements the Dynamic Programming method whereas ~~A~~ IDA* doesn't need dynamic programming.

IDA* is the memory constrained way of A*.

IDA* is better than A* in terms of path finding if there are no obstacles.
However, it requires more processing power than A* algorithm

Ans 3: Algorithm for Branch and Bound Search
- Push the root node to stack
- If the stack is empty, then stop and return failure.
- If the top node of stack is a goal node, then stop and return success.
- Else Pop the node from the stack. Process it and find all its successors. Find out the path containing all its successors as well as predecessors and then PUSH the successors which are belonging to the shortest path.
- Exit.

Algorithm for N-Workers & N-jobs allocation problem:-

- Initialize a dummy root node.
- Initialize a list of live nodes using min heap.
- Pop the node with minimum cost from the list.
- If ~~this~~ popped node is leaf node, print solution and exit.
- For each child of popped node, push the child to list.
- Mark the parent of pushed child as the popped node.
- Repeat from Step 3 until we reach a leaf node.