ASSIGNMENT-3

Artificial Intelligence

Page No.
Date

Q1 What is the difference between A* and AO* algorithm?

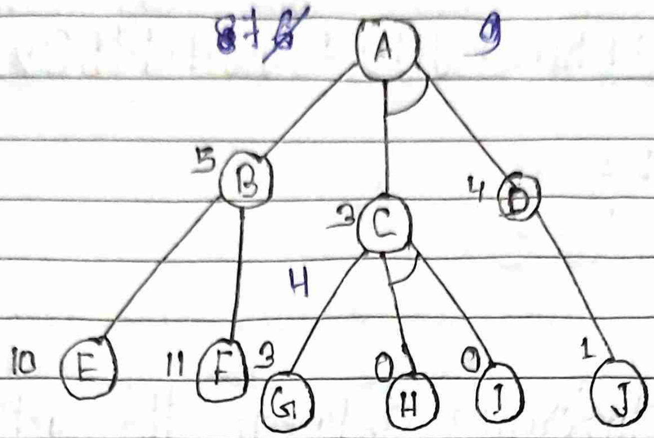
A* :- A* algorithm is an informed search algorithm leverages both the known cost to reach a point and a heuristic an estimate of the remaining distance to the goal.

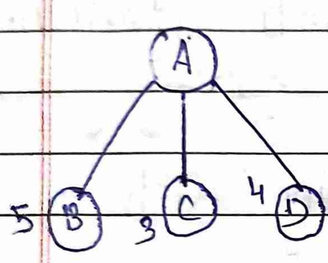
AO* :- The AO* algorithm is a variant of the A* algorithm, designed to be more adaptive and flexible, particularly in dynamic environments.

- Some key difference between A* and AO* algo.

A* Algorithm	AO* Algorithm
• Not designed for handling changes in the environment	Specifically designed to adapt to changes without initiating a new search.
• Uses the AND operation focusing on one path at a time.	Uses both OR and AND operations, exploring multiple paths simultaneously
• Generally more resource efficient, explores fewer nodes	May explore more nodes due to adaptability, potentially requiring more computational resources.

Q2 Apply AO^* algorithm on the graph





Step-1

Startie from node A, we first calculate the best Path

- $f(B) = g(B) + h(B)$
  $= 1 + 5 = 6$
- $f(C\text{-}D) = g(C) + h(C) + g(D) + h(D)$
  $= 1 + 3 + 1 + 4 = 9$

Step-2

- $f(B\text{-}E) = 1 + 10 = 11$
- $f(B\text{-}F) = 1 + 11 = 12$

- $f(A\text{-}B) = g(B) +$ updated $(h(B))$
  $= 1 + 6 \Rightarrow 7$

Step-3

- $f(C\text{-}G) = 1 + 3 = 4$
- $f(C\text{-}H\text{-}I) = 1 + 0 + 1 + 0 = 2$

• And finally the $f(A-C-D)$ needs to be updated

$f(A-C-D) = g(C) + h(C) + g(D) + \text{updated}(h(D))$.

$= 1 + 2 + 1 + 1 = 5$

Q3 Write the difference between the depth-first search and breath-first search algorithm.

DFS (Depth-First Search):-
Explores as far as possible along each branch before backtracking.

BFS (Breadth-First Search):-
Explores all neighbours of the persent depth level before moving to the next level.

DFS	BFS
• Used a stack LIFO	Uses a queue FIFO
• Goes deep into a branch first	Visits nodes level by level
• Requires less memory	Requires more memory
• May not find the shortest path in an unweighted graph	Always finds the shortest path in an unweighted graph

• May not visit all nodes in infinite or very deep graphs	Guaranteed to find a solution if one exists
• exm: Topological sorting maze solving, cycle detection	shortest path in unweighted graphs, level order traversal in tree.

Q4 What is the role of the heuristic function in A^* search?

The heuristic function ($h(n)$) plays a crucial role in the A^* search algorithm by helping it find the optimal and most efficient path.

Role of the Heuristic function in A^* search

1) Estimates the cost to the Goal:

- $h(n)$ Provides an estimate of the cost from the current node n ~~of~~ to the goal node.
- It does not need to be exact. Just an informed guess.

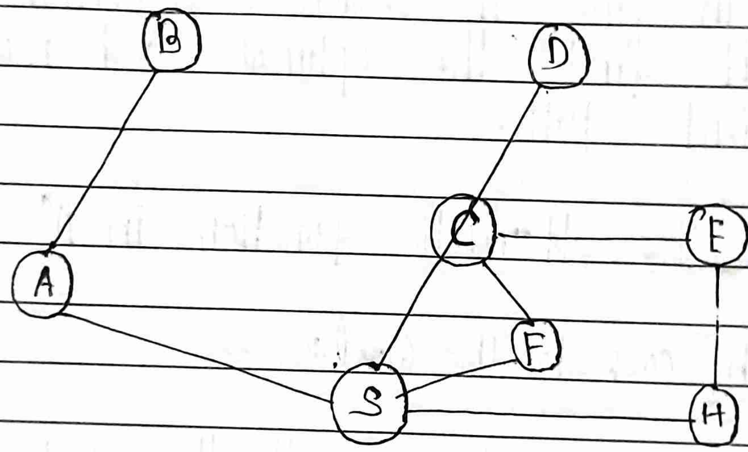
2) Guides the Search Efficiently:

- A^* uses $f(n) = g(n) + h(n)$
  - $g(n)$ = Cost from the start to node n.
  - $h(n)$ = estimated cost from n to the goal.

3) Reduces search Time :-
- A good heuristic prunes unnecessary paths, making the search faster.

4) Determines optimality :-
- If the heuristic is :
Admissible (never overestimates the true cost)
Consistent / monotonic (satisfies the triangle inequality).

Q5 Apply BFS and DFS on the following graph.



1. BFS Traversal (Level order using Queue)

Start from S

- Steps :

1. Start : S
2. Visit neighbours : A, C, F, H
3. From A → B
4. From C → D, E

Page No.
Date

5. From F → no new nodes
6. From H → (E already visited)
7. All nodes visited.

BFS order:.

S → A → C → F → H → B → D → E

2. DFS Traversal (Depth-Wise using stack)

Start from S.

Steps

1. Start : S
2. Choose neighbor : A
3. From A → B
4. Backtrack → A → S → next → C
5. From C → D → backtrack → E
6. Backtrack → C → S → next F
7. Backtrack → S →

DFS order :.

S → A → B → C → D → E → F → H

- BFS : S, A, C, F, H, B, D, E
- DFS : S, A, B, C, D, E, F, H