

Multiple Choice Questions for Lecture 4: Searching Algorithm in AI

Types of Search Algorithms

1. What are the two main types of search algorithms in AI?

- A) Fast search and slow search
- B) Easy search and hard search
- C) Uninformed search and informed search
- D) Manual search and automatic search

Answer: C) Uninformed search and informed search

2. What is another name for uninformed search?

- A) Smart search
- B) Blind algorithm
- C) Quick search
- D) Directed search

Answer: B) Blind algorithm

3. In uninformed search, what information does the agent have?

- A) Path cost and number of steps
- B) Only when it reaches a goal
- C) Distance to the goal
- D) Background information about the problem

Answer: B) Only when it reaches a goal

4. What information does informed search use?

- A) Only the path cost
- B) The exact steps to reach the goal
- C) Background information about the problem
- D) Only the branching factor

Answer: C) Background information about the problem

Uninformed Search Strategies

5. Breadth-First Search (BFS) explores:

- A) The deepest nodes first
- B) The lowest cost nodes first
- C) All nodes at the current depth before moving to the next level
- D) Nodes randomly

Answer: C) All nodes at the current depth before moving to the next level

6. What data structure does Breadth-First Search use?

- A) FIFO queue
- B) LIFO stack
- C) Priority queue
- D) Hashtable

Answer: A) FIFO queue

7. What does Uniform-Cost Search (UCS) prioritize?

- A) The closest node to the goal
- B) The lowest-cost path when multiple paths exist
- C) The deepest node in the tree
- D) Random nodes

Answer: B) The lowest-cost path when multiple paths exist

8. When does Uniform-Cost Search (UCS) act like Breadth-First Search (BFS)?

- A) When the goal is very far
- B) When paths have different costs
- C) When all transitions have equal costs
- D) When using a priority queue

Answer: C) When all transitions have equal costs

9. What data structure is used in Depth-First Search (DFS)?

- A) FIFO queue
- B) LIFO stack
- C) Priority queue
- D) Set

Answer: B) LIFO stack

10. Depth-First Search (DFS) explores:

- A) All nodes at the current level before moving deeper
- B) Branch nodes deeply before backtracking
- C) Random nodes
- D) Only the shortest paths

Answer: B) Branch nodes deeply before backtracking

11. What is Bidirectional Search?

- A) A search that goes in multiple random directions
- B) A search that runs from both start and goal states until they meet
- C) A search that only goes left and right
- D) A search that changes direction based on costs

Answer: B) A search that runs from both start and goal states until they meet

12. What is the main advantage of Bidirectional Search?

- A) It always finds the optimal solution
- B) It only needs to explore half the path compared to traditional methods
- C) It never gets stuck in loops
- D) It doesn't need any memory

Answer: B) It only needs to explore half the path compared to traditional methods

Informed Search Strategies

13. What function does Greedy Best-First Search use to evaluate nodes?

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

- B) $f(n) = g(n)$
- C) $f(n) = h(n)$
- D) $f(n) = 1/g(n)$

Answer: C) $f(n) = h(n)$

14. What does $h(n)$ represent in search algorithms?

- A) The height of the node
- B) The estimated cost from node n to the goal
- C) The path cost from the start to node n
- D) The number of children of node n

Answer: B) The estimated cost from node n to the goal

15. What is a limitation of Greedy Best-First Search?

- A) It's too slow
- B) It can get stuck in loops or dead ends
- C) It uses too much memory
- D) It's too complicated to implement

Answer: B) It can get stuck in loops or dead ends

16. What function does A^* search use to evaluate nodes?

- A) $f(n) = g(n) + h(n)$
- B) $f(n) = g(n)$
- C) $f(n) = h(n)$
- D) $f(n) = g(n) - h(n)$

Answer: A) $f(n) = g(n) + h(n)$

17. In A^* search, what does $g(n)$ represent?

- A) The goal state
- B) The path cost from the start to node n
- C) The estimated cost from node n to the goal

- D) The total search cost

Answer: B) The path cost from the start to node n

18. How does A* search combine UCS and Greedy Best-First Search?

- A) It alternates between using each algorithm
- B) UCS keeps solution cost low, Greedy Best-First helps find solutions quickly
- C) It runs both and takes the best result
- D) It only uses parts of each algorithm

Answer: B) UCS keeps solution cost low, Greedy Best-First helps find solutions quickly

Important Concepts

19. What is a "complete algorithm" in search?

- A) An algorithm that finds any solution
- B) An algorithm that guarantees finding a solution if one exists
- C) An algorithm that works in all situations
- D) An algorithm that is fully implemented

Answer: B) An algorithm that guarantees finding a solution if one exists

20. What is an "optimal solution" in search problems?

- A) The fastest solution
- B) The solution with the fewest steps
- C) The solution with the lowest cost
- D) The most recently found solution

Answer: C) The solution with the lowest cost