

Enrollment No. :
Centre Code :

Total number of printed pages = 4

2025

(MOOCs) IDC

Paper : MOOCIDC1013

Invigilator Sign.
Date :

(Artificial Intelligence: Search Methods
for Problem Solving)

Full Marks : 50

Pass Marks : 20

Time : 2 hours

(Question carries 1 mark each)

Please tick (✓) the correct answer

*Use of correction fluid is prohibited ** Multiple ticks will be treated as incorrect.*

-
1. In AI, problem-solving typically involves:
 - A. Random guessing
 - B. Searching through a problem space to find a solution
 - C. Memorizing facts
 - D. Only logical reasoning
 2. The "state space" refers to:
 - A. All possible states of a problem
 - B. Memory used by the computer
 - C. Programming language
 - D. Hardware resources
 3. A problem in AI can be defined by:
 - A. Initial state, actions, goal state, and path cost
 - B. Hardware only
 - C. Software only
 - D. Memory management
 4. A successor function in AI:
 - A. Generates possible next states from the current state
 - B. Evaluates goal achievement
 - C. Optimizes memory
 - D. Computes algorithm complexity
 5. Search methods in AI are mainly categorized as:
 - A. Logical and mathematical
 - B. Uninformed (blind) and informed (heuristic)
 - C. Manual and automated
 - D. Deterministic and non-deterministic
 6. Breadth-First Search (BFS) explores:
 - A. Deepest nodes first
 - B. Shallowest nodes first
 - C. Random nodes
 - D. Nodes with lowest cost
 7. Depth-First Search (DFS) explores:
 - A. Shallowest nodes first
 - B. Deepest nodes first
 - C. Nodes with least path cost
 - D. Random nodes
 8. Uniform-Cost Search expands:
 - A. Nodes based on depth
 - B. Nodes based on path cost
 - C. Random nodes
 - D. Only goal nodes
 9. Uninformed search does not use:
 - A. Heuristic information
 - B. Path cost
 - C. Node expansion
 - D. Goal test

10. BFS guarantees:
- A. Optimal solution if all step costs are equal
 - C. Non-optimal solution
11. Depth-limited search is a variant of:
- A. BFS
 - C. Uniform-cost search
12. Iterative Deepening Search (IDS) combines:
- A. BFS and DFS
 - C. Random and heuristic search
13. IDS is preferred because it:
- A. Requires less memory than BFS
 - C. Explores all states randomly
14. Depth-limited search may fail if:
- A. Goal is beyond the depth limit
 - C. BFS is used
15. IDS repeats node expansions but is still:
- A. Memory-efficient
 - C. Non-optimal
16. Heuristic search uses:
- A. Random guessing
 - C. Only BFS
17. A heuristic function ($h(n)$) estimates:
- A. Cost to reach goal from node n
 - C. Memory usage
18. Greedy Best-First Search chooses nodes based on:
- A. Path cost $g(n)$
 - C. Depth only
19. Greedy search is not always optimal because:
- A. It ignores path cost
 - C. It expands all nodes
20. A* search combines:
- A. Depth and breadth
 - C. Randomness and heuristic
21. Completeness of an algorithm means:
- A. It always finds a solution if one exists
 - C. It is fast
22. Optimality of an algorithm means:
- A. Finds the least-cost solution
 - C. Finds any solution
23. Time complexity of BFS is:
- A. $O(b^d)$
 - C. $O(b)$
24. Space complexity of DFS is:
- A. $O(d)$
 - C. $O(1)$
- B. Optimal solution always
 - D. Random solution
 - B. DFS with a depth limit
 - D. Best-first search
 - B. Uniform-cost and greedy search
 - D. Hill climbing and A* search
 - B. Guarantees optimality with reasonable memory
 - D. Avoids heuristics
 - B. Path cost is high
 - D. Heuristic is inaccurate
 - B. Time-efficient
 - D. Random
 - B. Knowledge about the problem to guide search
 - D. Only DFS
 - B. Depth of node
 - D. Number of children
 - B. Heuristic value $h(n)$
 - D. Random selection
 - B. It uses heuristics
 - D. It is memory-intensive
 - B. Path cost $g(n)$ and heuristic $h(n)$
 - D. BFS and DFS
 - B. It is memory-efficient
 - D. It finds multiple solutions
 - B. Uses minimum memory
 - D. Expands fewest nodes
 - B. $O(d)$
 - D. $O(1)$
 - B. $O(b^d)$
 - D. $O(b)$

25. Uniform-Cost Search is complete if:
- A. Step costs are positive
 - C. Infinite depth
26. Hill Climbing algorithm:
- A. Moves to neighbor with best value
 - C. Uses BFS
27. Steepest-Ascent Hill Climbing evaluates:
- A. One neighbor
 - C. Random nodes
28. Hill climbing can get stuck in:
- A. Local maxima
 - C. BFS tree
29. Simulated Annealing helps in:
- A. Escaping local maxima using probabilistic moves
 - C. DFS
30. Genetic algorithms use:
- A. Evolutionary principles like selection, crossover, and mutation
 - C. BFS
31. Minimax algorithm is used in:
- A. Single-agent search
 - C. Optimization problems
32. Alpha-Beta pruning:
- A. Reduces nodes evaluated in minimax without affecting optimality
 - C. Increases BFS efficiency
33. Evaluation function in games estimates:
- A. Utility or goodness of a game state
 - C. BFS depth
34. Game trees represent:
- A. All possible moves in a game
 - C. BFS expansion
35. Adversarial search assumes:
- A. Opponent acts optimally
 - C. Random moves
36. CSP involves:
- A. Finding values for variables satisfying constraints
 - C. BFS only
37. Examples of CSP include:
- A. Sudoku and map coloring
 - C. BFS
38. Backtracking search in CSP:
- A. Assigns variables and backtracks on failure
 - C. BFS
- B. Step costs can be zero
- D. Random nodes exist
- B. Explores all nodes
- D. Uses DFS
- B. All neighbors and chooses best
- D. Only goal nodes
- B. Optimal solution
- D. Path cost
- B. BFS
- D. Greedy search
- B. Heuristic functions only
- D. DFS
- B. Two-player games
- D. BFS only
- B. Finds heuristic value
- D. Estimates path cost
- B. Memory usage
- D. Number of successors
- B. Path cost only
- D. Random search
- B. Opponent is passive
- D. BFS only
- B. Random search
- D. DFS only
- B. Hill climbing
- D. DFS
- B. Random assignment
- D. DFS

39. Forward checking in CSP:
- A. Reduces domains of unassigned variables to prevent conflicts
 - C. BFS expansion
40. Arc consistency ensures:
- A. Every value in a variable's domain satisfies binary constraints
 - C. Heuristic optimization
41. Hill climbing is a type of:
- A. Local search
 - C. DFS
42. Informed search is also called:
- A. Blind search
 - C. DFS
43. Path cost in search is usually denoted by:
- A. $g(n)$
 - C. $f(n)$
44. Evaluation function in A* search:
- A. $f(n) = g(n) + h(n)$
 - C. $f(n) = h(n)$
45. Best-first search selects nodes based on:
- A. Lowest evaluation function value
 - C. Depth
46. Iterative deepening combines benefits of:
- A. BFS completeness and DFS memory efficiency
 - C. DFS speed only
47. Local maxima problem occurs in:
- A. Hill climbing
 - C. DFS
48. Random restart hill climbing:
- A. Reduces chance of getting stuck in local maxima
 - C. Uses BFS
49. Beam search keeps:
- A. Limited number of best nodes at each level
 - C. Only one node
50. Search methods in AI help in:
- A. Problem solving, planning, game playing, and optimization
 - C. Only DFS
- B. Chooses next variable randomly
- D. DFS expansion
- B. Random variable assignment
- D. BFS search
- B. BFS
- D. Uniform-Cost Search
- B. Heuristic search
- D. BFS
- B. $h(n)$
- D. $d(n)$
- B. $f(n) = g(n)$
- D. $f(n) = \text{depth}$
- B. Random selection
- D. BFS order
- B. BFS speed only
- D. Heuristic search only
- B. BFS
- D. Uniform-cost search
- B. Increases memory use
- D. Uses DFS
- B. All nodes
- D. None
- B. Only BFS
- D. Only heuristic evaluation