DPP 01

Data Science & Artificial Intelligence

Artificial Intelligence

DPP: 1

AI Foundation

Q1 Which of the following is an uninformed search strategy?
(A) A* Search
(B) Greedy Best-First Search
(C) Breadth-First Search (BFS)
(D) Hill Climbing

Q2 Depth-First Search (DFS) is particularly useful in situations where:
(A) The solution is not far from the root of the tree.
(B) There are a large number of nodes at deeper levels.
(C) We need to find the shortest path.
(D) The graph is complete and dense.

Q3 Which real-world application is most likely to use the A* search algorithm?
(A) Puzzle solving
(B) Scheduling tasks
(C) Sorting data
(D) Performing arithmetic operations

Q4 What distinguishes an informed search strategy from an uninformed one?
(A) The number of nodes generated
(B) The use of domain-specific knowledge or heuristics
(C) The data structure used
(D) The complexity of the algorithm

Q5 Which of the following algorithms is considered an informed search algorithm?
(A) Breadth-First Search (BFS)
(B) Depth-First Search (DFS)
(C) Uniform-Cost Search
(D) Greedy Best-First Search

Q6 What is a key characteristic of BFS?
(A) It always expands the least costly node.
(B) It explores one level of nodes at a time.
(C) It prioritizes nodes based on a heuristic.
(D) It is inherently recursive.

Q7 What is a disadvantage of using DFS in large search spaces?
(A) It can get stuck in loops.
(B) It always finds the shortest path.
(C) It is not complete.
(D) It cannot be implemented recursively.

Q8 Which algorithm is an example of uninformed search?
(A) A* Search
(B) Best-First Search
(C) Depth-Limited Search
(D) Iterative Deepening Depth-First Search

Q9 In the context of A* search, what does a heuristic function estimate?
(A) The exact cost from the start node to the goal
(B)

The remaining cost from the current node to the goal

(C) The number of nodes in the search space

(D) The total number of steps taken from the start

Q10 DFS is particularly suited for which kind of problem?

(A) Finding the shortest path in a weighted graph

(B) Maze solving where the path is not known in advance

(C) Problems where complete search space needs to be explored

(D) Real-time decision making in dynamic environments

Q11 What is the time complexity of Breadth-First Search for a graph with 'V' vertices and 'E' edges?

(A) O(V) (B) O(E)

(C) O(V + E) (D) O(VE)

Q12 Greedy best-first search primarily:

(A) Expands the node that is closest to the goal, according to a heuristic.

(B) Explores all possible paths simultaneously.

(C) Repeatedly explores the deepest node in the current frontier.

(D) Considers the cost of the path from the start node.

Q13 Is Depth-First Search (DFS) optimal?

(A) Yes, always

(B) No, never

(C) Yes, but only in finite spaces

(D) Yes, when the path cost is a non-decreasing function of the depth

Q14 BFS is often used in which real-world scenario?

(A) Optimizing travel routes in GPS navigation

(B) Building decision trees in machine learning

(C) Social networking applications to find connections

(D) Complex game theory computations

Q15 What is a common characteristic of informed search algorithms?

(A) They do not keep track of the state space.

(B) They use additional information to decide which node to explore next.

(C) They treat all nodes as equally likely to lead to a solution.

(D) They only use a FIFO queue for node storage.

Q16 What is a key difference between Greedy Best-First Search and A* Search?

(A) Greedy Best-First Search uses a heuristic, while A* Search does not.

(B) A* Search considers both the cost to reach the current node and an estimate to the goal.

(C) Greedy Best-First Search guarantees an optimal solution, while A* does not.

(D) A* is an uninformed search algorithm, whereas Greedy is informed.

Q17 What is the space complexity of Depth-First Search (DFS) in the worst case?

(A) O(log n) (B) O(n)

(C) O(n^2) (D) O(1)

Q18 Why might Breadth-First Search (BFS) be less practical in very large graphs?

(A) Due to its inability to find the shortest path.

(B) Because of its high space complexity.

(C) It is slower than DFS in all cases.

(D) BFS cannot handle cyclic graphs.

Q19 In what scenario would an informed search algorithm be less effective?

(A) When the heuristic closely approximates the true cost.
(B) When the problem space is large and complex.
(C) When there is no additional information or heuristic available.
(D) When searching for multiple goals simultaneously.

Q20 Depth-First Search is particularly effective in which type of application?
(A) Real-time strategy games.
(B) Pathfinding in road networks.
(C) Puzzle solving, like Sudoku.
(D) Database query optimization.

Q21 For A* search to be optimal, the heuristic used must be:
(A) Admissible, meaning it never overestimates the true cost.
(B) Consistent, meaning it decreases monotonically along a path.
(C) Both admissible and consistent.
(D) Neither admissible nor consistent.

Q22 Which search algorithm is better suited for searching cyclical graphs?
(A) Breadth-First Search (BFS)
(B) Depth-First Search (DFS)
(C) Both are equally suited
(D) Neither is suited

Q23 A major limitation of Greedy Best-First Search is:
(A) Its high space complexity.
(B) It is not guaranteed to find the shortest path.
(C) It cannot be used in large problem spaces.
(D) It only works with numeric data.

Q24 Which search algorithm adapts best to dynamic environments where the state space changes during the search?
(A) A*
(B) Iterative Deepening
(C) Dijkstra's Algorithm
(D) Real-Time A*

Q25 One advantage of Iterative Deepening Depth-First Search (IDDFS) over simple DFS is:
(A) Lower space complexity.
(B) It is faster in all cases.
(C) It combines the benefits of BFS's completeness and DFS's space efficiency.
(D) It works better in directed graphs.

Answer Key

Q1	(C)
Q2	(A)
Q3	(A)
Q4	(B)
Q5	(D)
Q6	(B)
Q7	(A)
Q8	(D)
Q9	(B)
Q10	(B)
Q11	(C)
Q12	(A)
Q13	(B)
Q14	(C)
Q15	(B)
Q16	(B)
Q17	(B)
Q18	(B)
Q19	(C)
Q20	(C)
Q21	(C)
Q22	(A)
Q23	(B)
Q24	(D)
Q25	(C)

Hints & Solutions

Q1 Text Solution:
BFS is an uninformed search strategy as it does not use any domain-specific information or heuristics to guide the search.

Q2 Text Solution:
DFS is efficient in scenarios where the solutions are likely to be at shallow depths because it explores as far as possible along a branch before backtracking.

Q3 Text Solution:
A* search is commonly used in pathfinding and graph traversal, which makes it ideal for applications like puzzle solving (e.g., solving a maze).

Q4 Text Solution:
Informed search strategies, unlike uninformed ones, use domain-specific knowledge or heuristics to make more efficient decisions about which paths to explore.

Q5 Text Solution:
Greedy Best-First Search is an informed search algorithm as it uses heuristics to prioritize nodes based on their estimated proximity to the goal.

Q6 Text Solution:
BFS is characterized by its level-by-level exploration, where all nodes at one depth are expanded before any nodes at the next level.

Q7 Text Solution:
DFS can get stuck exploring a path indefinitely (especially in graphs with cycles) due to its nature of going deep before backtracking.

Q8 Text Solution:
Iterative Deepening Depth-First Search is an uninformed search strategy because it doesn't use any domain-specific knowledge. It combines the depth-first search's space-efficiency and breadth-first search's completeness.

Q9 Text Solution:
The heuristic function in A* search estimates the cost to reach the goal from the current node. This estimation helps in guiding the search more efficiently toward the goal.

Q10 Text Solution:
DFS is well-suited for situations like maze solving, where the solution path is not known beforehand, and the algorithm can explore deeply with backtracking.

Q11 Text Solution:
The time complexity of BFS is O(V + E) because it explores each vertex and edge once.

Q12 Text Solution:
Greedy best-first search uses a heuristic to choose which node to expand by selecting the node that appears to be closest to the goal.

Q13 Text Solution:
DFS is not optimal because it may find a non-optimal solution if an optimal one exists at a shallower depth.

Q14 Text Solution:
BFS is used in social networks to find people within a certain number of steps from a

particular person, due to its level-by-level exploration nature.

Q15 Text Solution:
Informed search algorithms use additional information (heuristics) about the state space's structure to make more informed decisions about which nodes to explore.

Q16 Text Solution:
A* Search algorithm is more comprehensive than Greedy Best-First Search as it considers both the path cost up to the current node and the estimated cost from the current node to the goal.

Q17 Text Solution:
In the worst case, DFS can end up storing all the vertices in the stack if the path is very deep, leading to a space complexity of O(n).

Q18 Text Solution:
BFS can be less practical in very large graphs due to its high space complexity, as it needs to store a large number of nodes in the queue at any given time.

Q19 Text Solution:
Informed search algorithms rely on heuristics to guide the search. Without a reliable heuristic or additional information, their effectiveness diminishes.

Q20 Text Solution:
DFS is effective in applications like puzzle solving, where it can explore potential solutions deeply and backtrack as needed.

Q21 Text Solution:
For A* search to guarantee an optimal solution, the heuristic must be both admissible (never overestimates the true cost to reach the goal) and consistent (the estimated cost is always less than or equal to the estimated cost from any neighboring vertex to the goal plus the cost of reaching that neighbor).

Q22 Text Solution:
BFS is better suited for cyclical graphs as it is less likely to get trapped in cycles compared to DFS. BFS explores level by level, reducing the chances of indefinitely following a cycle.

Q23 Text Solution:
The major limitation of Greedy Best-First Search is that it is not guaranteed to find the shortest path because it always selects the node that appears to be closest to the goal, potentially overlooking shorter paths.

Q24 Text Solution:
Real-Time A* is designed to handle dynamic environments where the state space can change during the search. It makes decisions based on current information and adapts as new information becomes available.

Q25 Text Solution:
IDDFS combines the benefits of BFS's ability to find the shallowest solution (completeness) and DFS's space efficiency by iteratively deepening the depth limit.