DPP 01

Data Science & Artificial Intelligence

Artificial Intelligence

Searching Algorithm

Q1 Which of the following is a feature of uninformed search algorithms?
(A) They use a heuristic to guide the search.
(B) They have knowledge about the goal state's location.
(C) They treat all nodes equally and explore without any preference.
(D) They guarantee the optimal solution in all cases.

Q2 What is the main disadvantage of the Depth-First Search (DFS) algorithm in uninformed search?
(A) It cannot handle cycles.
(B) It is too slow.
(C) It can get stuck in loops or recursive paths.
(D) It cannot find a solution if one exists.

Q3 Which algorithm is guaranteed to find the shallowest solution if one exists?
(A) Greedy Best-First Search
(B) Depth-First Search
(C) Breadth-First Search
(D) Uniform Cost Search

Q4 In uniform-cost search, what determines the order of node expansion?
(A) The depth of the node
(B) The heuristic value of the node
(C) The path cost from the start node to the current node
(D) The estimated cost from the current node to the goal

Q5 What is the space complexity of Iterative Deepening Search (IDS)?
(A) O(b^d) (B) O(bm)
(C) O(d) (D) O(md)

Q6 What is the primary guiding factor for node selection in Greedy Best-First Search?
(A) The path cost from the start node
(B) The heuristic estimate of distance to the goal
(C) The depth of the node
(D) Random selection

Q7 What is a major drawback of Greedy Best-First Search?
(A) It is slow on large graphs.
(B) It does not guarantee the shortest path.
(C) It requires a lot of memory for small graphs.
(D) It cannot use heuristic functions.

Q8 In terms of space complexity, what is a potential issue with Greedy Best-First Search?
(A) It has a constant space requirement.
(B) It requires minimal space compared to other algorithms.
(C) It can consume a large amount of memory in the worst case.
(D) Its space complexity decreases with the size of the graph.

Q9 What type of data structure is commonly used to implement the open list in Greedy Best-First Search?

(A) Stack (B) Queue

(C) Priority Queue (D) Hash Table

Q10 Which scenario is most suitable for using Greedy Best-First Search?

(A) When the exact path to the goal is not important, and a quick solution is preferred.

(B) When finding the absolutely shortest path is crucial.

(C) When the search space is extremely large and memory is limited.

(D) When the heuristic often overestimates the distance to the goal.

Q11 What is the primary difference between Breadth-First Search (BFS) and Depth-First Search (DFS)?

(A) BFS uses a stack, while DFS uses a queue.

(B) BFS is optimal, while DFS is not.

(C) BFS can find the shortest path, while DFS cannot.

(D) BFS is slower than DFS.

Q12 In a graph with cycles, which mechanism is crucial for Depth-First Search (DFS) to avoid infinite loops?

(A) Limiting the depth of search

(B) Using a heuristic function

(C) Maintaining a visited list or set

(D) Alternating between BFS and DFS

Q13 Which statement is true about Uniform Cost Search (UCS)?

(A) UCS is identical to BFS when all step costs are equal.

(B) UCS cannot handle varying step costs.

(C) UCS prioritizes nodes based on their depth.

(D) UCS is a heuristic-based search.

Q14 What is the biggest limitation of uninformed search algorithms?

(A) They cannot find a solution in finite time.

(B) They require pre-knowledge about the goal state.

(C) They can be inefficient as they do not utilize domain-specific knowledge.

(D) They are only applicable to small search spaces.

Q15 How does Iterative Deepening Search (IDS) combine the advantages of BFS and DFS?

(A) By alternating between BFS and DFS in each iteration.

(B) By using a heuristic to guide its search.

(C) By repeatedly performing DFS with increasing depth limits.

(D) By randomly choosing between BFS and DFS strategies.

Q16 Which aspect of a problem is most crucial for the effectiveness of Greedy Best-First Search?

(A) The number of nodes in the graph

(B) The accuracy and suitability of the heuristic function

(C) The computational power available

(D) The presence of cycles in the graph

Q17 In Greedy Best-First Search, what happens if the heuristic is not admissible?

(A) The algorithm becomes faster.

(B) The algorithm guarantees to find the shortest path.

(C) The algorithm may overlook the optimal path.

(D) The heuristic becomes irrelevant to the search.

Q18 What is a potential drawback of using a very optimistic heuristic in Greedy Best-First Search?

(A) Increased time complexity

(B) Decreased space complexity
(C) Risk of missing shorter paths
(D) Guaranteed optimal solution

Q19 Which of the following scenarios is least likely to benefit from Greedy Best-First Search?
(A) When the exact shortest path is not critical
(B) When the heuristic is known to be very accurate
(C) When the goal is to find the absolutely shortest path
(D) When the search space is large but well-defined

Q20 How does Greedy Best-First Search differ from A* Search?
(A) Greedy Best-First Search uses only the heuristic, while A* uses both the heuristic and the cost so far.
(B) Greedy Best-First Search guarantees the shortest path, while A* does not.
(C) A* is an uninformed search algorithm, while Greedy Best-First Search is informed.
(D) A* uses a stack, while Greedy Best-First Search uses a queue.

Q21 What characteristic of a search problem makes Breadth-First Search (BFS) particularly memory-intensive?
(A) Deep search trees
(B) High branching factor
(C) Presence of cycles in the graph
(D) Large number of goal states

Q22 Which of the following best describes the Uniform Cost Search (UCS) when applied to a graph with uniform step costs?
(A) It behaves like Depth-First Search.
(B) It behaves like Breadth-First Search.
(C) It behaves like Greedy Best-First Search.
(D) It behaves uniquely and cannot be compared to other search algorithms.

Q23 During Depth-First Search (DFS), what is the maximum number of nodes that need to be stored in memory at any one time?
(A) Equal to the number of nodes in the graph
(B) Equal to the branching factor of the graph
(C) Equal to the depth of the deepest node
(D) Equal to the number of goal states

Q24 In which scenario would Iterative Deepening Search (IDS) be less efficient than standard Depth-First Search (DFS)?
(A) When the solution is very deep
(B) When the solution is near the root
(C) In graphs with high branching factors
(D) In graphs with many repeated states

Q25 In which scenario would Iterative Deepening Search (IDS) be less efficient than standard Depth-First Search (DFS)?
(A) When the solution is very deep
(B) When the solution is near the root
(C) In graphs with high branching factors
(D) In graphs with many repeated states

Q26 What type of graph makes Breadth-First Search (BFS) the least efficient in terms of space?
(A) Sparse graphs
(B) Graphs with uniform step costs
(C) Dense graphs
(D) Directed acyclic graphs (DAGs)

Q27 Which factor can significantly impact the performance of Greedy Best-First Search?
(A) The total number of nodes in the graph
(B) The number of goal states
(C) The accuracy of the heuristic function
(D)

The size of the data structure used to store nodes

Q28 What is a major limitation of Greedy Best-First Search in terms of completeness?
(A) It cannot find a solution even if one exists
(B) It always finds the least-cost solution
(C) It may fail to find a solution if the heuristic is misleading
(D) It requires a complete graph to work

Q29 In what situation would Greedy Best-First Search likely outperform A* Search?
(A) When the path cost is irrelevant and reaching the goal quickly is important
(B) When finding the absolute shortest path is essential
(C) In large search spaces with limited memory
(D) When the heuristic is not very accurate

Q30 How does Greedy Best-First Search typically fare in terms of optimality?
(A) It always finds the optimal solution
(B) It never finds the optimal solution
(C) It finds the optimal solution if the heuristic is perfect
(D) It may or may not find the optimal solution depending on the problem

Q31 Which of the following is true about the space complexity of Greedy Best-First Search?
(A) It is always constant
(B) It is always linear
(C) It can be high in large graphs with misleading heuristics
(D) It is irrelevant as the algorithm is space-efficient

Q32 In Depth-First Search (DFS), what happens when a node with no unvisited adjacent nodes is encountered?
(A) The algorithm restarts from the initial node.
(B) The algorithm backtracks to the nearest node with unvisited neighbors.
(C) The search switches to Breadth-First Search.
(D) The algorithm terminates immediately.

Q33 Which search algorithm expands the least number of nodes when the goal is the deepest node in the search tree?
(A) Breadth-First Search (BFS)
(B) Depth-First Search (DFS)
(C) Uniform Cost Search (UCS)
(D) Iterative Deepening Search (IDS)

Q34 What is the primary advantage of Uniform Cost Search (UCS) over Breadth-First Search (BFS)?
(A) UCS is faster in all cases.
(B) UCS is more memory efficient.
(C) UCS can handle varying step costs.
(D) UCS always uses a heuristic.

Q35 In which case does Breadth-First Search (BFS) perform poorly in terms of time complexity?
(A) When the goal node is close to the root
(B) When the search tree has a low branching factor
(C) When the goal node is at a very deep level
(D) When the graph has no cycles

Q36 What is the main limitation of using Iterative Deepening Search (IDS) in large search spaces?
(A) It is too memory-intensive.
(B) It may repeatedly explore the same nodes.
(C) It cannot handle cycles.
(D) It requires an accurate heuristic.

Q37 In which scenario can Greedy Best-First Search be most effective?
(A) When the heuristic is consistently overestimating distances
(B)

When the goal is to explore as many nodes as possible

(C) When the heuristic closely approximates the actual cost to the goal

(D) When the search space is small and simple

Q38 What is the primary reason Greedy Best-First Search might explore fewer nodes than A* Search?

(A) It always finds the shortest path.

(B) It prioritizes nodes based solely on heuristic, potentially leading to quicker goal discovery.

(C) It uses less memory.

(D) It cannot handle complex search spaces.

Q39 Which statement best reflects a limitation of Greedy Best-First Search in pathfinding problems?

(A) It cannot be used in grid-based maps.

(B) It always takes the longest path.

(C) It might not find the most efficient path if the heuristic is not ideal.

(D) It requires pre-knowledge of all paths.

Q40 How does the choice of heuristic impact the performance of Greedy Best-First Search?

(A) A less accurate heuristic always speeds up the search.

(B) An accurate heuristic ensures the shortest path is found.

(C) A poor heuristic can lead to unnecessary exploration.

(D) The heuristic has no impact on performance.

Q41 What happens when Greedy Best-First Search is used with a heuristic that always underestimates the true cost?

(A) It behaves exactly like Uniform Cost Search.

(B) It guarantees to find the shortest path.

(C) It might explore paths that appear closer to the goal but are not optimal.

(D) It becomes an uninformed search algorithm.

Q42 In Depth-First Search (DFS), what is the primary cause of its potential inefficiency in finding the shortest path?

(A) It prioritizes deeper nodes in the search tree.

(B) It uses too much memory for large graphs.

(C) It cannot handle cycles effectively.

(D) It requires an accurate heuristic.

Q43 What is a key characteristic of Uniform Cost Search (UCS) that distinguishes it from Breadth-First Search (BFS)?

(A) UCS is a depth-limited search.

(B) UCS uses a priority queue based on path cost.

(C) UCS cannot be used with weighted graphs.

(D) UCS is faster than BFS in all cases.

Q44 In which scenario would Breadth-First Search (BFS) be most appropriate?

(A) When the shortest path is required, and the search tree has a reasonable branching factor.

(B) When the search tree is very deep.

(C) When the search space is highly irregular.

(D) When the step costs are non-uniform.

Q45 Which search strategy is implemented by Iterative Deepening Search (IDS) to combine the advantages of DFS and BFS?

(A) It alternates between DFS and BFS in each iteration.

(B) It uses a heuristic to switch between DFS and BFS.

(C) It performs a DFS with a progressively increasing depth limit.

(D) It randomly selects between DFS and BFS strategies.

Q46

What is the main disadvantage of uninformed search strategies in large and complex search spaces?

(A) They require too much computational power.

(B) They are incapable of finding a solution.

(C) They are inefficient as they lack problem-specific knowledge.

(D) They cannot be implemented in modern programming languages.

Q47 What is the primary guiding principle of Greedy Best-First Search in choosing which node to expand next?

(A) The node with the lowest overall cost

(B) The node that is closest to the start point

(C) The node that appears to be closest to the goal based on a heuristic

(D) The node that has the fewest connections

Q48 What can be a major pitfall of using a poorly designed heuristic in Greedy Best-First Search?

(A) It makes the algorithm behave like BFS.

(B) It can lead the algorithm to get stuck in a loop.

(C) It can cause the algorithm to miss the goal entirely.

(D) It may cause the algorithm to explore less optimal paths.

Answer Key

Q1	(C)
Q2	(C)
Q3	(C)
Q4	(C)
Q5	(B)
Q6	(B)
Q7	(B)
Q8	(C)
Q9	(C)
Q10	(A)
Q11	(C)
Q12	(C)
Q13	(C)
Q14	(C)
Q15	(C)
Q16	(B)
Q17	(C)
Q18	(C)
Q19	(C)
Q20	(A)
Q21	(B)
Q22	(B)
Q23	(C)
Q24	(A)
Q25	(A)
Q26	(C)
Q27	(C)
Q28	(C)
Q29	(A)
Q30	(D)
Q31	(C)
Q32	(B)
Q33	(B)
Q34	(C)
Q35	(C)
Q36	(B)
Q37	(C)
Q38	(B)
Q39	(C)
Q40	(C)
Q41	(C)
Q42	(A)
Q43	(B)
Q44	(A)
Q45	(C)
Q46	(C)
Q47	(C)
Q48	(D)

Hints & Solutions

Q1 Text Solution:
Uninformed search algorithms do not have additional information about states or the goal; they explore the search space blindly.

Q2 Text Solution:
DFS can get trapped in loops or endlessly deep paths in search spaces with cycles or infinite depth.

Q3 Text Solution:
Breadth-First Search explores nodes level by level and is guaranteed to find the shallowest (or least-depth) solution first.

Q4 Text Solution:
Uniform-cost search expands nodes based on the cumulative cost from the start node, ensuring the lowest-cost path is explored first.

Q5 Text Solution:
The space complexity of IDS is O(bm), where b is the branching factor and m is the maximum depth, due to maintaining a stack of nodes at each depth level.
Greedy Best-First Search

Q6 Text Solution:
Greedy Best-First Search uses a heuristic estimate to the goal to guide its search, choosing the path that appears closest to the goal according to the heuristic.

Q7 Text Solution:
While Greedy Best-First Search can be efficient in reaching a goal, it does not guarantee the shortest or least-cost path due to its reliance on heuristic estimates.

Q8 Text Solution:
The space complexity can be large in the worst case as it may need to store all nodes, particularly if the heuristic leads it to explore many paths.

Q9 Text Solution:
A priority queue is typically used to manage the open list in Greedy Best-First Search, where nodes are prioritized based on their heuristic values.

Q10 Text Solution:
Greedy Best-First Search is suitable when a quick solution is more important than finding the shortest path, as it may not always lead to the optimal solution.

Q11 Text Solution:
BFS is guaranteed to find the shortest path in a graph, as it explores nodes level by level. DFS, on the other hand, dives deep into the graph, which may not lead to the shortest path.

Q12 Text Solution:
To avoid revisiting nodes and getting stuck in infinite loops, DFS must keep track of visited nodes, typically using a visited list or set.

Q13 Text Solution:
UCS becomes equivalent to BFS when all edges have the same cost, as it expands nodes in

order of their cumulative path cost.

Q14 Text Solution:
Uninformed search algorithms can be inefficient, especially in large search spaces, as they lack domain-specific information to guide the search process.

Q15 Text Solution:
IDS combines the space efficiency of DFS and the completeness and optimal path guarantee of BFS by incrementally deepening the depth limit for DFS.

Q16 Text Solution:
The effectiveness of Greedy Best-First Search heavily depends on the heuristic used; a good heuristic can lead to quick and effective solutions.

Q17 Text Solution:
If the heuristic is not admissible (i.e., it overestimates the cost), Greedy Best-First Search may not find the most optimal path, as it could be misled by the heuristic.

Q18 Text Solution:
An overly optimistic heuristic (underestimating the cost) may cause the algorithm to favor longer paths that appear to be shorter, leading to suboptimal solutions.

Q19 Text Solution:
Since Greedy Best-First Search does not guarantee the shortest path, it is not the best choice when the shortest path is the primary objective.

Q20 Text Solution:
Greedy Best-First Search is guided solely by the heuristic estimate to the goal, whereas A* combines this estimate with the cost from the start to the current node, balancing exploration and goal direction.

Q21 Text Solution:
BFS can be memory-intensive in problems with a high branching factor, as it needs to store all nodes at the current and next depth levels.

Q22 Text Solution:
In graphs with uniform step costs, UCS expands nodes in order of increasing path cost from the start node, effectively behaving like BFS.

Q23 Text Solution:
DFS requires storage for a path from the root to a leaf node, making its space requirement proportional to the maximum depth.

Q24 Text Solution:
IDS may become less efficient than DFS when the solution is very deep, as it redundantly explores shallow nodes multiple times.

Q25 Text Solution:
IDS may become less efficient than DFS when the solution is very deep, as it redundantly explores shallow nodes multiple times.

Q26 Text Solution:
BFS is least efficient in terms of space in dense graphs, where the high number of edges results in a large number of nodes at each level.

Q27 Text Solution:

The performance of Greedy Best-First Search heavily depends on the accuracy of the heuristic function. An inaccurate heuristic can lead the search away from the goal.

Q28 Text Solution:
Greedy Best-First Search may fail to find a solution if the heuristic function consistently misleads the search away from the goal.

Q29 Text Solution:
Greedy Best-First Search can be faster than A* in situations where quickly reaching the goal is more important than finding the least-cost path, as it aggressively follows the heuristic.

Q30 Text Solution:
The optimality of Greedy Best-First Search depends on the specific problem and the accuracy of the heuristic; it is not guaranteed to find the optimal solution.

Q31 Text Solution:
The space complexity of Greedy Best-First Search can be high, especially in large graphs where a misleading heuristic can cause the algorithm to explore many nodes.

Q32 Text Solution:
DFS backtracks to the most recent node that has unexplored paths when it reaches a dead end.

Q33 Text Solution:
DFS will expand the fewest nodes when the goal is the deepest, as it explores as deep as possible along each branch before backtracking.

Q34 Text Solution:
UCS's main advantage over BFS is its ability to handle varying step costs, expanding nodes based on their cumulative cost rather than their depth.

Q35 Text Solution:
BFS can be slow when the goal is at a deep level because it must explore all nodes at each level before reaching deeper levels.

Q36 Text Solution:
IDS's main limitation in large spaces is its redundancy; it explores shallow nodes multiple times in successive iterations.

Q37 Text Solution:
Greedy Best-First Search is most effective when the heuristic is accurate and closely approximates the true cost to the goal.

Q38 Text Solution:
Greedy Best-First Search can be quicker in reaching the goal by aggressively following the heuristic, sometimes at the expense of optimality.

Q39 Text Solution:
The limitation of Greedy Best-First Search in pathfinding is that it may not find the most efficient path if the heuristic does not accurately represent the true costs.

Q40 Text Solution:
The effectiveness of Greedy Best-First Search is heavily dependent on the heuristic; a poor heuristic can mislead the search, causing inefficient exploration.

Q41 Text Solution:

A heuristic that underestimates the cost can lead Greedy Best-First Search to favor paths that seem closer to the goal but may not be the most efficient overall.

Q42 Text Solution:

DFS can be inefficient for finding the shortest path as it explores as deep as possible in the search tree before backtracking, which may lead it away from closer solutions.

Q43 Text Solution:

The distinguishing feature of UCS is that it uses a priority queue where nodes are expanded based on the lowest cumulative path cost, unlike BFS which expands nodes level by level.

Q44 Text Solution:

BFS is most appropriate when finding the shortest path is essential, and the branching factor is not too high, as it explores uniformly across levels.

Q45 Text Solution:

IDS employs DFS with increasing depth limits, combining DFS's space efficiency and BFS's completeness and optimality.

Q46 Text Solution:

Uninformed search strategies can be inefficient in large and complex spaces because they do not utilize any problem-specific knowledge to guide the search.

Q47 Text Solution:

Greedy Best-First Search selects the node to expand next based on a heuristic that estimates how close a node is to the goal.

Q48 Text Solution:

A poorly designed heuristic in Greedy Best-First Search can mislead the search, causing it to explore paths that are less optimal or longer than necessary.