Topic: Solving problems as state spacesearch

Section: Multiple Choice Questions

1. In the context of state space search, what does a "state" primarily represent?

(a) A set of rules for moving between configurations

(b) A specific configuration of the problem at a given time

(c) The solution to the problem

(d) The cost of reaching the goal

2. Which of the following is NOT a fundamental component of a Production System for problem solving?

(a) A set of production rules

(b) Working memory

(c) A control strategy

(d) A pre-defined solution path

3. Breadth-First Search (BFS) is guaranteed to find the shortest path in an unweighted graph because:

(a) It explores all nodes at the current depth level before moving to the next level.

(b) It always chooses the path with the least number of nodes.

(c) It uses a heuristic function to guide its search.

(d) It prioritizes nodes based on their estimated cost to the goal.

4. A problem is considered "decomposable" if:

(a) The order of operators applied does not matter.

(b) It can be broken down into smaller, independent subproblems.

(c) The effects of operators are entirely predictable.

(d) It is possible to undo previous steps if they prove wrong.

5. What is a major disadvantage of Depth-First Search (DFS) in state space exploration?

(a) It is not guaranteed to find a solution even if one exists.

(b) It requires excessive memory for storing visited nodes.

(c) It can get trapped in infinite loops or very long paths without finding a solution.

(d) It is generally slower than Breadth-First Search for deep solutions.

6. Which search algorithm is an example of an "informed search" technique?

(a) Breadth-First Search

(b) Depth-First Search

(c) Best-First Search

(d) Uniform Cost Search

7. Hill Climbing is a greedy search algorithm that:

(a) Explores all possible paths to find the optimal solution.

(b) Always moves to the state that appears best from the current state.

(c) Backtracks when it reaches a local optimum.

(d) Guarantees finding the global optimum for any problem.

8. An "operator" or "production rule" in state space search defines:

(a) The target state that the search aims to reach.

(b) How to transition from one state to another.

(c) The initial configuration of the problem.

(d) The evaluation function used to rank states.

9. What is the primary purpose of a "heuristic function" in search algorithms?

(a) To guarantee completeness of the search.

(b) To reduce the search space by guiding the algorithm towards promising states.

(c) To ensure optimality of the found solution.

(d) To determine the exact cost of a path from the start to the goal.

10. Which of the following is a characteristic of a problem where "undoing previous steps" is straightforward and not costly, making backtracking a viable strategy?

(a) Decomposable problem

(b) Ignorable problem

(c) Predictable problem

(d) Absolute importance problem

11. Compared to Breadth-First Search, Depth-First Search typically requires:

(a) More memory for shallow solutions.

(b) Less memory, as it only needs to store the current path.

(c) Equal memory, regardless of solution depth.

(d) Memory proportional to the total number of nodes in the state space.

12. In the context of Best-First Search, how does the algorithm decide which node to expand next?

(a) It expands the node that was most recently added to the fringe.

(b) It expands the node that has the lowest estimated cost to the goal, according to a heuristic function.

(c) It expands the node that is closest to the initial state.

(d) It expands nodes randomly until the goal is found.

13. What is the main issue with Hill Climbing getting stuck on a "local optimum"?

(a) It cannot find any solution from that point.

(b) It might find a suboptimal solution instead of the globally optimal one.

(c) It consumes too much memory at that point.

(d) It indicates that the problem has no solution.

14. For a heuristic function h(n) to be "admissible", it must satisfy which condition?

(a) h(n) >= actual cost from n to goal

(b) h(n) = 0 for all nodes n

(c) h(n) <= actual cost from n to goal

(d) h(n) must always be a large positive number

15. If a search problem has a very large and deep state space, but solutions are known to be at a relatively shallow depth, which uninformed search algorithm might be more suitable in terms of finding a solution quickly (though not necessarily optimal)?

(a) Uniform Cost Search

(b) Depth-First Search (with depth limit)

(c) Breadth-First Search

(d) Iterative Deepening Depth-First Search

Answers

1. (b)

2. (d)

3. (a)

4. (b)

5. (c)

6. (c)

7. (b)

8. (b)

9. (b)

10. (b)

11. (b)

12. (b)

13. (b)

14. (c)

15. (b)

Topic: Production system

Section: Multiple Choice Questions

16. Which of the following best describes the core purpose of a production system in AI?

a) To execute predefined functions sequentially.

b) To model problem-solving using a set of rules and a global database.

c) To generate random solutions and test their validity.

d) To store large amounts of unstructured data.

17. A typical production system consists of which of the following three main components?

a) Input/Output devices, CPU, Memory.

b) Working Memory, Production Rules, Control Strategy.

c) Sensor, Actuator, Knowledge Base.

d) Data Structures, Algorithms, User Interface.

18. In a production system, what is the primary function of the "working memory"?

a) To store the history of all executed rules.

b) To hold the current state of the problem and relevant facts.

c) To manage the conflict resolution set.

d) To define the initial and goal states of the problem.

19. The "control strategy" (or interpreter) in a production system is responsible for:

a) Updating the production rules dynamically.

b) Determining which rule to apply next from the conflict set.

c) Generating new problem states without rules.

d) Translating user queries into system commands.

20. The recognize-act cycle is the fundamental operational process of a production system. Which step immediately follows the "act" phase?

a) Match

b) Select

c) Execute

d) Termination check

21. What are "production rules" typically composed of in a production system?

a) If-Then statements linking conditions to actions.

b) Complex mathematical equations.

c) Binary code sequences for logic gates.

d) User interface specifications.

22. "Conflict resolution" is a crucial part of a production system when:

a) No rules can be applied to the current state.

b) Multiple rules are matched by the working memory at the same time.

c) The goal state has been reached.

d) The working memory becomes empty.

23. One of the primary advantages of using a production system for problem solving is its:

a) Guaranteed optimal solution for all problems.

b) High computational efficiency for large state spaces.

c) Modularity and ease of adding/modifying knowledge.

d) Elimination of the need for a search algorithm.

24. A common limitation of simple production systems, especially for complex problems, is:

a) Its inability to represent knowledge declaratively.

b) The difficulty in modifying existing rules.

c) Potential for inefficient search due to non-optimal rule selection.

d) Lack of support for backward chaining reasoning.

25. How do production rules relate to state transitions in the context of state-space search?

a) Production rules define the initial and goal states.

b) Production rules represent the operators that transform one state into another.

c) Production rules determine the search algorithm (e.g., DFS or BFS).

d) Production rules are only used for heuristic evaluation, not state changes.

26. Which type of problem is most suited for a production system approach?

a) Highly numerical problems requiring complex calculations.

b) Problems that can be broken down into a set of condition-action rules.

c) Problems requiring real-time, continuous data processing.

d) Problems with a fixed, small number of possible states.

27. Consider a simple game where rules dictate moves. How would a production system typically choose the next move compared to a brute-force Breadth-First Search (BFS)?

a) A production system would always explore moves layer by layer like BFS.

b) A production system would apply rules based on current conditions, potentially leading directly to the goal, while BFS explores systematically.

c) A production system would only evaluate moves based on a heuristic function, unlike BFS.

d) A production system cannot be used for game-playing problems.

28. Which of the following is generally NOT considered a core characteristic or component of a basic production system?

a) A set of production rules.

b) A working memory representing the current state.

c) A conflict resolution strategy.

d) A pre-defined, fixed search depth limit.

29. If a production system lacks a robust conflict resolution strategy, what is a likely consequence when multiple rules are applicable?

a) The system will halt, indicating an error.

b) It might enter an infinite loop or make arbitrary, potentially non-optimal choices.

c) All matched rules will execute simultaneously.

d) It will automatically prioritize rules with fewer conditions.

30. In the context of problem-solving using a production system, when is a problem considered "solved"?

a) When all production rules have been executed at least once.

b) When the working memory becomes completely empty.

c) When a goal state or condition specified by a rule is reached.

d) When the system has exhausted all possible rule applications.

Answers

16. (b)

17. (b)

18. (b)

19. (b)

20. (d)

21. (a)

22. (b)

23. (c)

24. (c)

25. (b)

26. (b)

27. (b)

28. (d)

29. (b)

30. (c)

Topic: Problem characteristics

Section: Multiple Choice Questions

31. Which problem characteristic describes whether the problem can be broken down into smaller, independent subproblems that can be solved separately?

(a) Predictability

(b) Decomposability

(c) Recoverability

(d) Solution Optimality

32. Consider a problem where a step, once taken, cannot be undone or retracted without significant cost or effort. This characteristic refers to:

(a) Irrecoverability

(b) Predictability

(c) Relative solution

(d) Finite universe

33. In the context of problem characteristics, if the outcome of an action is always known with certainty, the problem is said to have a:

(a) Consistent knowledge base

(b) Recoverable solution

(c) Predictable universe

(d) Decomposable structure

34. A problem like playing Chess is typically characterized by:

(a) A decomposable problem

(b) An unpredictable universe

(c) An absolute solution

(d) An irrecoverable nature (in terms of moves)

35. Which characteristic addresses whether the goal is to find any satisfactory solution or the best possible solution?

(a) Decomposability

(b) Solution optimality (absolute vs. relative)

(c) Consistency of knowledge base

(d) Problem scope

36. An 8-puzzle problem, where a wrong move can be undone by simply making another move to revert the state, demonstrates which characteristic?

(a) Irrecoverable

(b) Predictable

(c) Recoverable

(d) Non-decomposable

37. When solving a problem using a production system, if the rules' applicability depends on dynamic environmental factors that change unexpectedly, it relates to the problem's:

(a) Decomposability

(b) Recoverability

(c) Predictability of the universe

(d) Solution optimality

38. Problems like symbolic integration, where the main problem can be broken down into integrating smaller terms, are examples of:

(a) Irrecoverable problems

(b) Problems with an unpredictable universe

(c) Decomposable problems

(d) Problems with a relative solution

39. If an AI agent needs to navigate a maze where walls might randomly appear or disappear, this affects the problem's:

(a) Decomposability

(b) Recoverability

(c) Predictability

(d) Solution type

40. In the context of state space search, if a problem is irrecoverable, what implication does it have for algorithms like Depth-First Search?

(a) Backtracking is highly efficient.

(b) Incorrect branches can lead to dead ends requiring restarts or significant effort.

(c) Heuristic functions become irrelevant.

(d) The problem becomes inherently decomposable.

41. Which of the following problems most likely has an 'absolute' solution?

(a) Finding the quickest route from city A to city B using current traffic data.

(b) Designing a satisfactory aesthetic painting.

(c) Playing a game of Tic-Tac-Toe to win.

(d) Recommending a movie based on user preferences.

42. The characteristic of a problem where the available knowledge for solving it is complete and does not change during the problem-solving process is called:

(a) Dynamic knowledge

(b) Consistent knowledge base

(c) Heuristic knowledge

(d) Imperfect knowledge

43. A problem requiring real-time interaction with an external, uncertain environment (e.g., robotic control in a busy factory floor) is often characterized by:

(a) A predictable universe and decomposable nature.

(b) An unpredictable universe and a need for real-time decisions.

(c) An absolute solution and a consistent knowledge base.

(d) A recoverable nature and static knowledge.

44. When a problem has multiple satisfactory solutions, but no single "best" solution can be definitively determined, it falls under the characteristic of:

(a) Absolute solution

(b) Unique solution

(c) Relative solution

(d) Optimal solution

45. Which problem characteristic would significantly impact the effectiveness of a heuristic function that relies on estimating the cost to reach a goal, assuming the environment is constantly changing in unknown ways?

(a) Decomposability

(b) Recoverability

(c) Predictability of the universe

(d) Consistency of knowledge base

Answers

31. (b)

32. (a)

33. (c)

34. (d)

35. (b)

36. (c)

37. (c)

38. (c)

39. (c)

40. (b)

41. (c)

42. (b)

43. (b)

44. (c)

45. (c)

Topic: Depth First Search

Section: Multiple Choice Questions

46. Which data structure is primarily used by Depth First Search (DFS) to keep track of nodes to be visited?

(a) Queue

(b) Stack

(c) Hash Table

(d) Priority Queue

47. In the context of state space search, Depth First Search (DFS) explores:

(a) All nodes at the current level before moving to the next level.

(b) As far as possible along each branch before backtracking.

(c) Nodes based on an estimated cost to the goal.

(d) Nodes randomly until the goal is found.

48. Is Depth First Search (DFS) guaranteed to find a solution if one exists in a finite state space, assuming no cycles or redundant paths are explored?

(a) Yes, it is complete.

(b) No, it is not complete.

(c) Only if the solution is at a shallow depth.

(d) Only if the state space is a tree.

49. When considering search optimality, Depth First Search (DFS) is generally considered:

(a) Optimal, as it always finds the shortest path.

(b) Not optimal, as it may find a longer path to the goal.

(c) Optimal, but only in unweighted graphs.

(d) Optimal, but only if used with a heuristic function.

50. What is the space complexity of Depth First Search (DFS) in the worst case, in terms of the maximum depth of the search tree 'm' (representing the maximum size of the search stack)?

(a) O(bm)

(b) O(b^m)

(c) O(m)

(d) O(b+m)

51. What is the time complexity of Depth First Search (DFS) in the worst case, where 'b' is the branching factor and 'm' is the maximum depth of the search tree?

(a) O(b)

(b) O(m)

(c) O(b^m)

(d) O(bm)

52. Compared to Breadth-First Search (BFS), a key advantage of Depth First Search (DFS) in very large search spaces can be its:

(a) Guarantee of finding the optimal solution.

(b) Lower time complexity.

(c) Lower memory requirement.

(d) Ability to avoid getting stuck in infinite paths.

53. A major drawback of standard Depth First Search (DFS) without modification is its potential to:

(a) Fail to find a solution even if one exists, if it goes down an infinitely deep path.

(b) Require excessive memory for shallow solutions.

(c) Always find the longest path to the goal.

(d) Be slower than Breadth-First Search on average.

54. Which of the following problems is generally well-suited for a Depth First Search approach?

(a) Finding the shortest path in an unweighted graph.

(b) Checking if a graph is connected.

(c) Finding all paths between two nodes.

(d) Finding a solution in a state space with a known shallow goal and a high branching factor.

55. Depth-Limited Search (DLS) is a variation of DFS that:

(a) Explores nodes in increasing order of their heuristic values.

(b) Explores nodes up to a specified maximum depth 'L'.

(c) Explores all nodes at one level before moving to the next.

(d) Uses a priority queue to select the next node.

56. Iterative Deepening Depth-First Search (IDDFS) combines the advantages of DFS and BFS by being:

(a) Optimal in solution cost but incomplete.

(b) Complete and optimal, but with high memory usage.

(c) Complete and optimal, with relatively low memory usage.

(d) Incomplete and sub-optimal, but very fast.

57. When searching a state space, if it is known that solutions are likely to be found at great depths and the branching factor is very large, which uninformed search strategy might be a practical choice due to memory constraints?

(a) Breadth-First Search

(b) Uniform Cost Search

(c) Depth-First Search

(d) Bidirectional Search

58. Depth First Search (DFS) is classified as an uninformed search algorithm because it:

(a) Requires knowledge about the goal state to estimate costs.

(b) Does not use any problem-specific knowledge beyond the state space structure.

(c) Always finds the optimal path to the goal.

(d) Uses a heuristic function to guide its search.

59. In the context of a state space search problem like the 8-puzzle, a DFS algorithm would typically explore:

(a) All possible moves from the current state before exploring any of their resulting states.

(b) One sequence of moves as far as possible until it reaches a goal or a dead end, then backtracks.

(c) Moves that are estimated to bring it closer to the goal state first.

(d) Moves randomly to explore the state space.

60. How do production rules relate to Depth First Search in a state space search problem?

(a) Production rules define the heuristic function used by DFS.

(b) DFS uses production rules to prune branches of the search tree.

(c) Production rules define the legal actions that transform one state into another, generating successor nodes for DFS.

(d) DFS determines the order in which production rules are applied to find the optimal sequence.

Answers

46. (b)

47. (b)

48. (a)

49. (b)

50. (c)

51. (c)

52. (c)

53. (a)

54. (b)

55. (b)

56. (c)

57. (c)

58. (b)

59. (b)

60. (c)

Topic: Breadth-First Search

Section: Multiple Choice Questions

61. What data structure is primarily used to implement Breadth-First Search (BFS) to manage the nodes to be explored?

(a) Stack

(b) Queue

(c) Hash Map

(d) Priority Queue

62. Is Breadth-First Search (BFS) complete?

(a) Yes, if the branching factor is finite and the search space is finite.

(b) No, it can get stuck in infinite loops.

(c) Only if the graph is acyclic.

(d) Only if a heuristic function is used.

63. Under what condition is Breadth-First Search (BFS) guaranteed to find an optimal solution (shortest path in terms of number of edges)?

(a) When the graph is unweighted and all edge costs are uniform (e.g., 1).

(b) When a suitable heuristic function is available.

(c) When the search space is very deep.

(d) Only for acyclic graphs.

64. What is the time complexity of Breadth-First Search (BFS) on a graph with V vertices and E edges, assuming an adjacency list representation?

(a) O(V^2)

(b) O(V + E)

(c) O(E log V)

(d) O(V log V)

65. What is the worst-case space complexity of Breadth-First Search (BFS) in terms of branching factor 'b' and depth of the shallowest solution 'd'?

(a) O(b multiplied by d)

(b) O(b raised to the power of d)

(c) O(d)

(d) O(b)

66. Compared to Depth-First Search (DFS), what is a primary characteristic of Breadth-First Search (BFS) in exploring a state space?

(a) BFS explores as deep as possible before backtracking.

(b) BFS explores all nodes at the current depth level before moving to the next depth level.

(c) BFS is generally more memory-efficient than DFS.

(d) BFS always finds a non-optimal solution first.

67. In a problem where the goal is to find the path with the minimum number of moves in an unweighted grid (e.g., finding the shortest path for a robot), which search algorithm is most appropriate?

(a) Depth-First Search

(b) Breadth-First Search

(c) Hill Climbing

(d) Iterative Deepening Depth-First Search (IDDFS)

68. When applying Breadth-First Search (BFS) to a state space search problem, what does each "level" of the search tree primarily represent?

(a) The accumulated cost of reaching that state.

(b) The number of steps (or moves) taken from the initial state.

(c) The heuristic value of the state.

(d) The total number of nodes visited so far.

69. For problems where the solution is known to be relatively shallow and finding the optimal path length in an unweighted graph is crucial, which search strategy is generally preferred?

(a) Depth-First Search

(b) Breadth-First Search

(c) Greedy Best-First Search

(d) A* Search

70. Which of the following statements correctly differentiates Breadth-First Search (BFS) from informed search algorithms like Best-First Search?

(a) BFS uses a heuristic function to guide its search, while informed search does not.

(b) BFS explores nodes based on their estimated cost to the goal, while informed search explores level by level.

(c) BFS is an uninformed search algorithm, exploring all equally promising paths, while informed search uses problem-specific knowledge.

(d) BFS is guaranteed to be optimal for all graphs, whereas informed search might not be.

71. In Breadth-First Search, the "frontier" or "open list" contains:

(a) Nodes that have already been visited and expanded.

(b) Nodes that have been generated but not yet expanded.

(c) Nodes that have not yet been generated.

(d) The path from the start node to the current node.

72. To prevent Breadth-First Search (BFS) from entering infinite loops or performing redundant computations in graphs with cycles, what mechanism is typically used?

(a) Limiting the maximum search depth.

(b) Maintaining a "visited" set or list of expanded nodes.

(c) Using a priority queue to order node expansion.

(d) Employing a heuristic function to detect cycles.

73. Consider a problem described by a production system. If Breadth-First Search is used, what determines the order in which new states (generated by applying production rules) are added to the queue?

(a) Their estimated cost to the goal state.

(b) The order in which the production rules are defined.

(c) They are added to the end of the queue, maintaining a FIFO order.

(d) They are added to the front of the queue, similar to a stack.

74. What is a significant practical limitation of Breadth-First Search (BFS) in very large state spaces, especially when the solution is deep?

(a) It may not find the shortest path in unweighted graphs.

(b) It is susceptible to getting stuck in infinite loops even with a visited set.

(c) Its memory requirements can become prohibitive due to storing all nodes at the current level.

(d) It requires a precise heuristic function, which is not always available.

75. Breadth-First Search (BFS) can be considered a special case of which other search algorithm?

(a) Depth-First Search (DFS)

(b) Hill Climbing

(c) Uniform Cost Search (UCS), where all edge costs are uniform (e.g., 1).

(d) Greedy Best-First Search

Answers

61. (b)

62. (a)

63. (a)

64. (b)

65. (b)

66. (b)

67. (b)

68. (b)

69. (b)

70. (c)

71. (b)

72. (b)

73. (c)

74. (c)

75. (c)

Topic: Heuristic function

Section: Multiple Choice Questions

76. Which of the following best describes the primary purpose of a heuristic function in state space search?

(a) To guarantee finding the shortest path to the goal.

(b) To estimate the cost from the current state to the goal state.

(c) To enumerate all possible states in the search space.

(d) To determine the exact sequence of actions to reach the goal.

77. In the context of Best-First Search, how does a heuristic function guide the search process?

(a) It prioritizes nodes based on their depth in the search tree.

(b) It expands the node with the lowest estimated cost to the goal.

(c) It always explores the left-most child node first.

(d) It randomly selects the next node to expand.

78. An admissible heuristic function is one that:

(a) Never overestimates the cost to reach the goal.

(b) Always underestimates the cost to reach the goal.

(c) Provides the exact cost to reach the goal.

(d) Never underestimates the cost to reach the goal.

79. Consider the 8-puzzle problem. Which of the following is a common heuristic function used to estimate the cost to the goal state?

(a) Number of nodes visited so far.

(b) Sum of Manhattan distances of each tile from its goal position.

(c) Depth of the current state in the search tree.

(d) Number of total possible moves from the current state.

80. Which type of search algorithm primarily relies on a heuristic function to guide its exploration, potentially at the cost of completeness or optimality?

(a) Breadth-First Search

(b) Depth-First Search

(c) Hill Climbing

(d) Uniform Cost Search

81. A heuristic function h(n) is said to be consistent if, for every node n and every successor n' of n, with step cost c(n, n'), the following condition holds:

(a) h(n) <= h(n')

(b) h(n) <= c(n, n') + h(n')

(c) h(n) >= c(n, n') + h(n')

(d) h(n) = c(n, n') + h(n')

82. In an informed search algorithm like A*, if the heuristic function h(n) always returns 0 for all states, which uninformed search algorithm does A* degenerate into?

(a) Depth-First Search

(b) Breadth-First Search

(c) Uniform Cost Search

(d) Iterative Deepening Depth-First Search

83. The quality of a heuristic function directly impacts:

(a) Only the completeness of a search algorithm.

(b) Only the optimality of a search algorithm.

(c) Both the efficiency and optimality of a search algorithm.

(d) Neither the efficiency nor the optimality of a search algorithm.

84. Which of the following is a characteristic of an "uninformed" search strategy?

(a) It uses problem-specific knowledge to guide the search.

(b) It does not use any domain-specific heuristic information.

(c) It guarantees finding the optimal solution.

(d) It always explores the deepest nodes first.

85. For a heuristic to be truly effective in practical applications, it should be:

(a) Complex to calculate but highly accurate.

(b) Simple to calculate and reasonably accurate.

(c) Always overestimate the cost to the goal.

(d) Randomly generated for each search instance.

86. If a heuristic function consistently overestimates the true cost to the goal, what is a potential consequence for an algorithm like A*?

(a) It guarantees to find the optimal path more quickly.

(b) It may fail to find the optimal path.

(c) It will always explore fewer nodes.

(d) It will behave identically to Breadth-First Search.

87. Which of the following is NOT a direct benefit of using a good heuristic function in state space search?

(a) Reduced search time.

(b) Reduced memory usage.

(c) Guaranteed finding of a solution if one exists.

(d) Improved efficiency in finding optimal or near-optimal solutions.

88. The Euclidean distance between the current state and the goal state is a common heuristic used for which type of problem?

(a) Solving Sudoku puzzles.

(b) Finding the shortest path in a 2D grid navigation problem.

(c) Determining the optimal move in a chess game.

(d) Rearranging tiles in the N-puzzle problem.

89. What is the relationship between an admissible heuristic and a consistent heuristic?

(a) All admissible heuristics are consistent.

(b) All consistent heuristics are admissible.

(c) Admissible and consistent heuristics are unrelated.

(d) A heuristic can be admissible but never consistent.

90. What happens if a heuristic function for Hill Climbing returns a local optimum that is not the global optimum?

(a) The algorithm will restart from the initial state.

(b) The algorithm will continue searching for a better solution indefinitely.

(c) The algorithm will terminate, having found a sub-optimal solution.

(d) The algorithm will automatically backtrack to explore other paths.

Answers

76. (b)

77. (b)

78. (a)

79. (b)

80. (c)

81. (b)

82. (c)

83. (c)

84. (b)

85. (b)

86. (b)

87. (c)

88. (b)

89. (b)

90. (c)

Topic: Hill climbing

Section: Multiple Choice Questions

91. Which of the following best describes the primary characteristic of a Hill Climbing search algorithm?

a) It explores all possible paths systematically before making a decision.

b) It always guarantees finding the global optimum solution.

c) It makes decisions based only on the current state and its immediate neighbors.

d) It requires a deep understanding of the entire problem space beforehand.

92. A significant limitation of the Simple Hill Climbing algorithm is its tendency to get stuck in:

a) Optimal solutions

b) Unexplored regions

c) Local maxima

d) Global minima

93. In Steepest-Ascent Hill Climbing, how is the next state chosen from the current state's neighbors?

a) The first neighbor encountered that is better than the current state.

b) A random neighbor that is better than the current state.

c) The neighbor that offers the greatest improvement in heuristic value.

d) The neighbor that has the lowest heuristic value.

94. Compared to Breadth-First Search (BFS), Hill Climbing is generally considered to be:

a) More complete and optimal.

b) More memory-intensive due to storing all explored nodes.

c) Less memory-intensive as it only keeps track of the current state.

d) Guaranteed to find the shortest path.

95. What is the role of a heuristic function in a Hill Climbing algorithm?

a) To determine if the current state is the goal state.

b) To generate all possible successor states.

c) To evaluate the "goodness" or promise of a state in reaching the goal.

d) To manage the memory usage of the search process.

96. Define the term "local maxima" in the context of Hill Climbing search.

97. Explain the fundamental difference between Simple Hill Climbing and Steepest-Ascent Hill Climbing in terms of successor selection.

98. Discuss one major advantage and one major disadvantage of Hill Climbing when compared to uninformed search strategies like Depth-First Search (DFS) or Breadth-First Search (BFS).

99. Describe a "plateau" in the search space for a Hill Climbing algorithm and explain why it poses a problem for this search technique.

100. In what scenarios might Hill Climbing be a preferred search strategy over more exhaustive methods, despite its limitations? Provide one practical reason.

101. Hill Climbing search terminates when:

a) It has explored all possible states in the search space.

b) It reaches a state from which no neighboring state is better.

c) A predefined time limit is exceeded.

d) The heuristic function returns zero.

102. Briefly explain one common strategy used to try and overcome the problem of getting stuck in local maxima in Hill Climbing.

103. What is Stochastic Hill Climbing? How does it differ from Simple Hill Climbing in choosing the next state?

104. Why is Hill Climbing often referred to as a "greedy" algorithm?

105. Consider a state space search problem formulated as a production system. How does the "production rules" component influence the state transitions within a Hill Climbing algorithm?

Answers

91. (c)

92. (c)

93. (c)

94. (c)

95. (c)

96. (textual answer expected)

97. (textual answer expected)

98. (textual answer expected)

99. (textual answer expected)

100. (textual answer expected)

101. (b)

102. (textual answer expected)

103. (textual answer expected)

104. (textual answer expected)

105. (textual answer expected)

Topic: Best First Search

Section: Multiple Choice Questions

106. Best First Search uses which of the following to guide its search?

(a) FIFO queue

(b) Heuristic function

(c) LIFO stack

(d) Random selection

107. Which data structure is typically used to implement the OPEN list in Best First Search?

(a) Queue

(b) Stack

(c) Priority Queue

(d) Hash Table

108. Best First Search is considered an example of an:

(a) Uninformed search algorithm

(b) Informed search algorithm

(c) Adversarial search algorithm

(d) Local search algorithm

109. A primary disadvantage of Best First Search, especially without a well-chosen heuristic, is its potential to:

(a) Not find any solution

(b) Explore the entire state space

(c) Get stuck in local optima

(d) Be computationally expensive due to large open lists

110. In the context of state space search, what is a production system?

(a) A system for manufacturing goods

(b) A set of rules and a control strategy for changing states

(c) A method for storing problem states efficiently

(d) A framework for evaluating heuristic functions

111. Which problem characteristic indicates whether an algorithm can unwind its steps easily to backtrack?

(a) Decomposability

(b) Ignorable

(c) Recoverable

(d) Predictable

112. True or False: Best First Search always finds the optimal path to the goal.

113. What is the primary data structure used to maintain the 'OPEN' list in Best First Search?

114. Fill in the blank: Best First Search is classified as an ___________ search algorithm because it uses problem-specific knowledge to guide its search.

115. State one significant advantage of using Best First Search over uninformed search algorithms like Breadth-First Search.

116. True or False: Hill Climbing algorithm is guaranteed to find the global optimum.

117. What does it mean for a search algorithm to be 'complete'?

118. Describe the main difference in node selection between Best First Search and Depth-First Search.

119. Fill in the blank: In a production system, a set of rules is applied to change the state from an initial state to a ___________ state.

120. Name two characteristics of problems that influence the choice of a search algorithm.

Answers

106. (b)

107. (c)

108. (b)

109. (d)

110. (b)

111. (c)

112. False

113. Priority Queue

114. Informed

115. Efficiency / Faster convergence / Goal-directed (any one)

116. False

117. It will always find a solution if one exists.

118. Best First Search selects node with best heuristic value; Depth-First Search selects the deepest unexpanded node.

119. Goal

120. Decomposable, Ignorable, Recoverable, Predictable (any two)