# Multiple-Choice Questions

1. In the context of the Greedy Best-First Search algorithm, why is the path from Arad through Sibiu and Fagaras to Bucharest not optimal despite reaching the goal?

A) Because it avoids unnecessary nodes

B) Because it is the shortest possible path

C) Because it does not use heuristic functions

D) Because there is a shorter path through Rimnicu Vilcea and Pitesti

\*\*Answer: D) Because there is a shorter path through Rimnicu Vilcea and Pitesti\*\*

2. What does the heuristic function \( h(n) \) represent in the context of informed search algorithms?

A) The exact cost from the node to the goal

B) The estimated cost from the node to the goal

C) The number of nodes expanded

D) The depth of the node in the search tree

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

3. In adversarial search, what is the primary purpose of the Minimax algorithm?

A) To minimize the search space

B) To maximize the loss for the opponent

C) To choose the best move assuming the opponent also plays optimally

D) To simulate the game with random moves

\*\*Answer: C) To choose the best move assuming the opponent also plays optimally\*\*

# True/False Questions

1. The \( A^\* \) search algorithm is optimal if the heuristic function is admissible.

\*\*Answer: True. Explanation: An admissible heuristic never overestimates the true cost to reach the goal, ensuring that \( A^\* \) is both complete and optimal when used with such a heuristic.\*\*

2. In Greedy Best-First Search, expanding the node closest to the goal guarantees finding the shortest path to the goal.

\*\*Answer: False. Explanation: Greedy Best-First Search does not always find the shortest path as it focuses only on the node closest to the goal, which can lead to suboptimal paths.\*\*

3. Genetic Algorithms are based on the process of natural selection and typically operate on binary strings.

\*\*Answer: True. Explanation: Genetic Algorithms simulate the process of natural selection using operations like crossover and mutation on a population of solutions, often represented as binary strings.\*\*

4. Local Search Algorithms always maintain a search tree to find the solution.

\*\*Answer: False. Explanation: Local Search Algorithms do not maintain a search tree; instead, they operate using a single solution which is iteratively improved upon.\*\*

5. Hill-Climbing Search can sometimes get stuck at a local maximum and fail to find the global maximum.

\*\*Answer: True. Explanation: Hill-Climbing Search makes greedy choices by moving to the highest neighboring value, which can trap it at local maxima, away from the global maximum.\*\*

# Short Answer Questions

1. Explain the significance of the branching factor and depth in evaluating the complexity of the Minimax algorithm.

\*\*Answer: The branching factor, represented as \( b \), indicates the average number of moves available at each point, and the depth, \( m \), indicates the maximum number of levels in the game tree. The complexity of Minimax is \( O(b^m) \), making both the branching factor and depth crucial for understanding the algorithm's computational demands, especially in games with large search spaces like chess.\*\*

2. Describe how Alpha-Beta pruning enhances the Minimax algorithm.

\*\*Answer: Alpha-Beta pruning enhances Minimax by reducing the number of nodes evaluated in the search tree. It does this by eliminating branches that cannot possibly affect the final decision, based on the best currently explored options. This pruning allows the algorithm to search deeper into the game tree with the same computational resources, thereby increasing efficiency without compromising the outcome of Minimax.\*\*