Mock Test

Topic: daa
Difficulty: Advanced
Total Questions: 5
Time Allowed: 10 minutes

Instructions:

1. Attempt all questions
2. Each question carries equal marks
3. Time allowed: 10 minutes

1. Prove or disprove: A greedy algorithm always yields an optimal solution for the 0/1 knapsack problem.

A) Disprove; a greedy approach may yield a suboptimal solution.
B) Prove; a greedy approach always finds the optimal solution.
C) Disprove; the problem is NP-hard and no known polynomial-time algorithm guarantees optimality.
D) Prove; a greedy approach is optimal if item values are inversely proportional to their weights.

2. Describe a scenario where using a Fibonacci heap would significantly outperform a binary heap in Dijkstra's algorithm. Justify your answer by comparing their time complexities in relevant operations.

A) Sparse graphs with a large number of edges.
B) Dense graphs with a small number of vertices.
C) Graphs with a highly skewed degree distribution, where many vertices have a small degree and few have a very large degree.
D) All graph types would see similar performance.

3. Analyze the time complexity of finding the kth smallest element in an unsorted array using Quickselect. Consider both average and worst-case scenarios.

A) Average: O(n), Worst-case: O(n^2)
B) Average: O(log n), Worst-case: O(n)
C) Average: O(n^2), Worst-case: O(n)
D) Average: O(n), Worst-case: O(n log n)

4. Explain how to modify the standard merge sort algorithm to efficiently find the number of inversions in an array. What is the resulting time complexity?

A) O(n log n)
B) O(n)
C) O(n^2)
D) O(log n)

5. Design an efficient algorithm to find the longest common subsequence (LCS) of three sequences. Explain its time and space complexity.

A) Time: O(n^3), Space: O(n^3)
B) Time: O(n^2), Space: O(n^2)
C) Time: O(n log n), Space: O(n)
D) Time: O(n), Space: O(1)

Answer Key

1. Correct Answer: A
Explanation: The 0/1 knapsack problem is known to be NP-hard. Greedy algorithms often provide a good approximation but do not guarantee an optimal solution for all instances. A counter-example easily disproves option B.

2. Correct Answer: C
Explanation: Fibonacci heaps excel when there are many decrease-key operations. In Dijkstra's algorithm using a Fibonacci heap, decrease-key operations are amortized O(1), compared to O(log n) for binary heaps. A highly skewed distribution results in many decrease-key operations, thus favoring Fibonacci heaps.

3. Correct Answer: A
Explanation: Quickselect's average-case time complexity is linear, O(n). However, in the worst-case scenario (e.g., consistently selecting the smallest or largest pivot), it can degrade to O(n^2).

4. Correct Answer: A
Explanation: By modifying the merge step of merge sort to count inversions (disordered pairs) during the merging process, the total number of inversions can be found within the same O(n log n) time complexity as the standard merge sort.

5. Correct Answer: A
Explanation: A dynamic programming approach can solve the LCS problem for three sequences. The straightforward extension of the two-sequence algorithm results in a time complexity of O(n^3) and space complexity of O(n^3), where n is the length of the sequences (assuming roughly equal lengths).