Mock Test

Topic: Gate
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. Consider a 2-D array A[m][n] where m and n are large. You need to find the largest sum of any submatrix within A. What algorithm would be most efficient for solving this problem, considering both time and space complexity, and what is its time complexity?

A) Kadane's Algorithm with a modified approach: O(m*n*min(m,n))
B) Divide and Conquer approach: O(m*n*log(m)*log(n))
C) Dynamic Programming: O(m*n)
D) Brute-force approach: O(m^3*n^3)

2. Design a data structure to efficiently support the following operations on a set of integers: (1) Insert(x): insert integer x, (2) Delete(x): delete integer x, (3) RandomElement(): return a random element from the set with uniform probability. What is the amortized time complexity of these operations?

A) Insert: O(1), Delete: O(n), RandomElement: O(1)
B) Insert: O(log n), Delete: O(log n), RandomElement: O(1)
C) Insert: O(1), Delete: O(1), RandomElement: O(1)
D) Insert: O(log n), Delete: O(log n), RandomElement: O(log n)

3. You are given a directed acyclic graph (DAG) representing tasks and their dependencies. You need to find the longest path in the DAG. Describe an efficient algorithm and its time complexity.

A) Dijkstra's Algorithm: O(E log V)
B) Bellman-Ford Algorithm: O(VE)
C) Topological Sort followed by Dynamic Programming: O(V+E)
D) Floyd-Warshall Algorithm: O(V^3)

4. What is the minimum number of comparisons required in the worst case to find the second smallest element in a set of n unsorted elements?

A) n - 1
B) n + " (n) - 2
C) n + log n
D) 3n/2

5. Given a binary tree, design an algorithm to check if it's a complete binary tree. What is the time complexity of your algorithm?

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

Answer Key

1. Correct Answer: C
Explanation: Dynamic programming provides an efficient solution by building a sum matrix. Kadane's algorithm is suitable for 1-D arrays. Divide and conquer and brute-force are less efficient for this specific problem.

2. Correct Answer: C
Explanation: A hash table with an array for storage and a separate array to track the number of occupied slots allows for O(1) amortized time for all three operations. Balanced trees achieve O(log n) but not O(1) for RandomElement.

3. Correct Answer: C
Explanation: Topological sort orders the vertices, then dynamic programming efficiently calculates the longest path lengths from the sorted order.

4. Correct Answer: A
Explanation: Find the smallest element (n-1 comparisons), then find the smallest element among the remaining (n-2) elements (n-2 comparisons) in the worst case, total n-1 comparisons

5. Correct Answer: B
Explanation: A level-order traversal can efficiently check for completeness by verifying if all levels except possibly the last are completely filled and nodes are filled from left to right. This is achievable in O(n) time.