Roll No. .... **Total Pages : 03**

BT-5/D-23 45274

DESIGN AND ANALYSIS OF ALGORITHMS
PC-CS-AIML-303A

Time : Three Hours] [Maximum Marks : 75

Note : Attempt *Five* questions in all, selecting at least *one* question from each Unit. All questions carry equal marks.

Unit I

1. (a) Explain the difference between an algorithm's time complexity and space complexity and why are these metrics important in algorithm analysis ? Discuss the time complexity of quick sort.

(b) How do you determine the time and space complexity of algorithms using recurrence relations and what are the key steps in the analysis ?

2. (a) How do you determine the time complexity of an algorithm using asymptotic notation when analyzing loops, recursion and nested operations ?

(b) How does the substitution method work in solving recurrence relations, and under what circumstances is it most effective ?

Unit II

3. (a) How is dynamic programming applied to longest common subsequence problem ?

   (b) How does the Fibonacci heap perform insertions and decrease-key operations while maintaining its unique structure and properties ?

4. (a) Discuss the algorithm for performing a decrease-key operation in a binomial heap and its time complexity.

   (b) How does a greedy algorithm handle problems involving combinatorial optimization, such as the traveling salesman problem or the knapsack problem ?

Unit III

5. (a) What is the difference between topological sorting and topological ordering of a graph, and how are they used ?

   (b) What is the fundamental principle behind the Bellman-Ford algorithm ? What is the time complexity of the Bellman-Ford algorithm and how does it compare to other shortest path algorithms like Dijkstra's algorithm ?

6. (a) Describe the concept of NP-completeness and the implications it has for algorithm design and problem-solving ?

(b) How does Kruskal's algorithm work to find the minimum spanning tree and what is its time complexity ?

Unit IV

7. (a) How does the bitonic sorting network handle cases where the input list is not a power of 2 and what techniques are used to adapt the network for such scenarios ?

(b) Provide a step-by-step explanation of the Ford-Fulkerson algorithm, particularly in finding the maximum flow in a network.

8. (a) How is the maximum bipartite matching problem represented and formulated as a graph problem and what are the key elements involved ?

(b) What is the concept of the "min-cut" in network flow problems, and how is it related to the maximum flow found by the Ford-Fulkerson method ? Discuss.