Registration No.: | | | | | | | | | | |

Total Number of Pages: 02 **Course: IDD (B.Tech and M.Tech)**
Sub_Code: CSPC2006

4th Semester Regular Examination: 2024-25
SUBJECT: Design and Analysis of Algorithms
BRANCH(S): CE, CSE, CSEAI, CSEAIML, CSEDS, CSIT, CST, ELECTRICAL & C.E, IT
Time: 3 Hours
Max Marks: 100
Q.Code: S429
Answer Question No.1 (Part-I) which is compulsory, any eight from Part-II and any two from Part-III.
The figures in the right hand margin indicate marks.

Part-I

Q1 **Answer the following questions:** **(2 x 10)**

a) Solve the recurrence: T(n) = 2T(n/2) + n using Master Theorem.
b) Write the recurrence relation for binary search.
c) What is the best case time complexity of insertion sort?
d) When is merge sort preferred over quicksort?
e) What is the difference between Dijkstra's and Bellman-Ford algorithms?
f) Define the 0/1 Knapsack problem.
g) Differentiate between Greedy and Dynamic Programming approaches.
h) What is a state space tree?
i) Give an example where backtracking is not an efficient approach.
j) What is a reducibility in the context of NP problems?

Part-II

Q2 **Only Focused-Short Answer Type Questions- (Answer Any Eight out of Twelve)** **(6 x 8)**

a) Explain and compare Big O, Θ, and Ω notations with suitable examples.
b) Analyze the time complexity of a recursive factorial function.
c) Explain AVL trees and their rotations with an example.
d) Describe disjoint set operations and their implementation using union by rank and path compression.
e) Describe the Matrix Chain Multiplication problem and its dynamic programming solution.
f) Explain Prim's and Kruskal's algorithms with example graphs.
g) Solve the Longest Common Subsequence problem using dynamic programming for two given strings.
h) Explain the Rabin-Karp algorithm with an example.
i) Describe the working of the Knuth-Morris-Pratt (KMP) algorithm.
j) Solve the 8-Queens problem using backtracking.

k) Define NP-completeness and show that Subset Sum is NP-complete.

l) Describe the approximation algorithm for the vertex cover problem.

Part-III

Only Long Answer Type Questions (Answer Any Two out of Four) **(16 x 2)**

Q3 Discuss in detail the concepts of P, NP, NP-complete, and NP-hard classes. Explain with suitable examples and diagrams. Include the significance of polynomial-time reductions. **(16)**

Q4 Explain the implementation and complexity analysis of Red-Black Trees. Compare with AVL Trees. **(16)**

Q5 Compare Greedy and Dynamic Programming approaches using the 0/1 Knapsack problem. Provide detailed pseudocode and complexity analysis. **(16)**

Q6 Discuss various string matching algorithms (Naive, Rabin-Karp, Finite Automata, and KMP) and compare their time complexities. **(16)**