RAMAKRISHNA MISSION VIDYAMANDIRA

NEP Syllabus B.Sc. Computer Science Honours

Semester-IV

Course Code: 4CMSMJC1

Course Type: Major Course

Course Outcome:

i) Recall the fundamental concepts of sorting and searching techniques.
ii) Understand the growth of functions, asymptotic notations, and methodologies for analyzing algorithms with different examples.
iii) Solve computational problems using different algorithm design techniques, graph algorithms, Network flow algorithm, string matching algorithms, randomized algorithm.
iv) Analyze the correctness and efficiency of these algorithms.
v) Critically evaluate different algorithmic design strategies to determine the best approach for given computational problem.
vi) Design innovative algorithmic solutions for complex problems using advanced methodologies.

4CMSMJC1: Design and Analysis of Algorithms

Credit: 3 **Marks: 50**

Foundations of Algorithms:
Requirements and Methodologies for Analysing Algorithms, Growth of Functions, Asymptotic Notations and their Properties: Big-Oh Notation, Big-Theta Notation, Big-Omega Notation, Little-Oh Notation, Little-Omega Notation, Proof of Correctness, Case Studies of Sorting and Searching Techniques, Amortized Analysis: Aggregate Analysis, Accounting Method, Potential Method. **[8L]**

Recurrences: Recurrence Relations, Generating Functions, Linear Recurrence Relations with Constant Coefficients and their Solution by Method of Iteration, Characteristic Roots and Generating Functions, Recursion Tree and Master Method.
[8L]

Algorithm Design Techniques:
Basic Concepts and Case Studies of Divide and Conquer (Strassen's Matrix Multiplication), Dynamic Programming (Matrix Chain Multiplication, 0-1 Knapsack Problem), Greedy algorithm (Fractional Knapsack Problem), Backtracking (N-Queen's Problem), Branch-and-Bound (Job Scheduling Problem).
[8L]

Graph Representations and Algorithms: Adjacency Matrix Representation, Adjacency List Representation, Graph Traversal Algorithms: Breadth-First Search, Depth-First Search, Minimum

Spanning Tree Algorithms: Prim's Algorithm, Kruskal's Algorithm, Shortest Path Algorithms: Bellman Ford Algorithm, Dijkstra's Algorithm, Floyd-Warshall Algorithm. **[7L]**

Network Flow Algorithms:
Introduction to Flow Network and Cut, Finding Maximum Flow, Ford-Fulkerson Method. **[3L]**

String Matching Algorithms:
Basic Concepts and Terminologies, Naive Algorithm, Knuth-Morris-Pratt Algorithm. **[3L]**

Randomized Algorithms:
Basic Concepts and Allied Theories, Randomized Quick Sort. **[3L]**

Computational Complexity:
Basic Concepts and Terminologies, P, NP, NP-Hard and NP-Complete Classes, Their Relation and Case Study, Satisfiability Problem, Reducibility. **[5L]**

4CMSMJC1: Design and Analysis of Algorithms Laboratory

Credit: 1 **Marks: 25**

Implementing Problems on Divide and Conquer, Dynamic Programming, Greedy Approach, Backtracking, Branch and Bound. **[12L]**

Implementation of Graph Algorithms: BFS, DFS, Prim's Algorithm, Kruskal's Algorithm, Bellman-Ford Algorithm, Dijkstra's Algorithm, Floyd-Warshall Algorithm. **[16 L]**

Implementation of String Matching Algorithms: Knuth-Morris-Pratt Algorithm. **[2L]**

Recommended Books:

1. Introduction to Algorithms by Cormen et al, 3rd Edition, PHI Learning.
2. Algorithm Design by Kleinberg, Tardos;Pearson.
3. Algorithms & Data structure by Ellis Horowitz, H.Sahani, Sanguthevar Rajasekaran, 2nd Edition; Universities Press.
4. The Art of Programming (Vol.1 and Vol.2) by Donald. E. Knuth, 3rd Edition, Pearson.
5. Computer Algorithms: Introduction to Design and Analysis by Sara Baase, Van Gelder; 1st Edition; Pearson.