RAMAKRISHNA MISSION VIDYAMANDIRA

NEP Syllabus B.Sc. Computer Science Honours

Semester-VII

Course Code: 7CMSMJC2

Course Type: Major Course

Course Outcome:

- Recall the fundamental terminologies, methodologies and asymptotic notations used for measuring the efficiency and correctness of algorithms.
- Explain recurrence relations, amortized analysis and the principal algorithm-design paradigms such as divide-and-conquer, dynamic programming, greedy, backtracking and branch-and-bound.
- Apply advanced graph, network-flow, string-matching and number-theoretic algorithms to model and solve real-world computational problems.
- Analyse the time and space complexity of randomized, parallel and approximation algorithms and compare alternative algorithmic solutions to the same problem.
- Evaluate the computational complexity of problems and classify them into P, NP, NP-Hard and NP-Complete classes using reducibility and lower-bound arguments.
- Design advanced data structures such as heaps (binary, binomial and Fibonacci), balanced search trees and disjoint-set structures for efficient storage and retrieval.
- Develop optimised solutions to advanced computing problems by selecting and integrating suitable data structures with appropriate algorithmic strategies.

7CMSMJC2: Advanced Algorithms and Data Structures

Credit: 3 **Marks: 50**

Review of Foundations of Algorithms: **[5L]**

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. **[2L]**

Recurrence Relation, Solving Recurrence Relation using Substitution Method, Recursion Tree and Master Method. **[1L]**

Amortized Analysis: Aggregate Analysis, Accounting Method, Potential Method, Proof of Correctness. **[2L]**

Review of Algorithm Design Techniques and Graph Algorithms: Basic Concepts and Case Studies of Divide and Conquer, Dynamic Programming, Greedy Algorithm, Backtracking, Branch-and-Bound. Graph 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. **[8L]**

Network Flow Algorithms: Introduction to Flow Network and Cut, Finding Maximum Flow, Ford-Fulkerson Method. **[2L]**

Lower Bound Theory: Concept of Lower Bound and Decision Trees, Lower Bounds for Bubble Sort, Binary Search. **[2L]**

Randomized Algorithms: Basic Concepts and Allied Theories, Randomized Quick Sort, Randomized Find. **[2L]**

String Matching Algorithms: Basic Concepts and Terminologies, Naive Algorithm, Knuth-Morris-Pratt Algorithm. **[2L]**

Polynomials and Number Theoretic Algorithms: Introduction, Representation of Polynomial, Polynomial Arithmetic, Fast Fourier Transform, Modular Arithmetic, Solving Modular Linear Equations, Chinese Remainder Theorem. **[3L]**

Computational Complexity: Basic Concepts and Terminologies, P, NP, NP-Hard and NP-Complete Classes, Reducibility, Case Study (Satisfiability Problems, Graph Coloring Problem). **[2L]**

Approximation Algorithms: Basic Concepts, Approximation Algorithm for Vertex Cover Problem and Travelling Salesman Problem. **[2L]**

Parallel Algorithms: Need for Parallel Algorithms, Models of Computation, Analysing Parallel Algorithms, Parallel Algorithms for Sorting and Searching. **[3L]**

Review of Linear and Non-Linear Data Structures: Vector, List, Sequence ADT: Array, Stack, Queue, Linked List; Different Sorting and Searching Algorithms; Binary Tree, Binary Search Tree, Threaded Binary Tree; Height Balanced Tree: AVL Tree, B-Tree, and B*-Tree; Hashing. **[4L]**

Advanced Data Structures:

Heap: Binary Heap, Binomial Heap, Fibonacci Heap. **[2L]**

Huffman Coding, Red-Black Tree. **[2L]**

Sets and Disjoint Set Data Structure. **[1L]**

7CMSMJC3 (Practical): Advanced Algorithms and Data Structures Laboratory

Credit: **Marks: 25**

Implementation and empirical run-time comparison of standard sorting and searching algorithms to verify their asymptotic behaviour, Implementation of divide-and-conquer, dynamic-programming, greedy, backtracking and branch-and-bound techniques, Implementation of graph traversal, minimum-spanning-tree, shortest-path algorithms Implementation of the Ford-Fulkerson method for computing the maximum flow in a network, Implementation of randomized algorithms (randomized quick sort and randomized select), Implementation of string-matching algorithms (naive and Knuth-Morris-Pratt), Implementation of number-theoretic algorithms (modular exponentiation, the extended Euclidean algorithm and the Chinese Remainder Theorem) and polynomial multiplication using the Fast Fourier Transform, Implementation and operations on advanced data structures (binary heap, binomial heap, Fibonacci heap, AVL tree, Red-Black tree and the disjoint-set/union-find structure), Mini-project applying a suitable combination of advanced data structures and algorithmic techniques to a real-world problem. **[30L]**

Texts / References

1. Introduction to Algorithms by T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, 3rd Edition, PHI Learning.
2. Algorithm Design by K. Tardos, Pearson.
3. Randomized Algorithms by R. Motwani and P. Raghavan, Cambridge University Press.
4. Parallel Computing: Theory & Practice by Michael J. Quinn, Second Edition, Tata McGraw Hill.
5. Parallel Computers: Architecture and Programming by V. Rajaraman and C Siva Ram Murthy, PHI Learning.
6. The Design and Analysis of Parallel Algorithms by S. G. Akl, Prentice Hall.
7. Data Structures using C by A. S. Tanaenbaum, Y. Langram, and M. J. Augestein, Pearson Education, 2004.
8. Fundamentals of Data Structures in C++ by E. Horowitz, S. Sahni and Dinesh Mehta, University Press, 2007.
9. Data Structures, Algorithms and Applications in C++ by S. Sahni, University Press (India) Pvt. Ltd, 2nd edition, Universities Press Orient Longman Pvt. Ltd.
10. Data Structures and Algorithms in C++ by M. T. Goodrich, R. Tamassia, and Mount, Wiley Student Edition, John Wiley and Sons.
11. Data Structures and Algorithm Analysis in C++ by Mark Allen Weiss, Pearson Education Ltd., Second Edition.
12. Data Structures and Algorithms in C++, Third Edition by A. Drozdek, Thomson.