

OBJECTIVES:

- To understand the usage of algorithms in computing.
- To learn and use hierarchical data structures and its operations
- To learn the usage of graphs and its applications.
- To select and design data structures and algorithms that is appropriate for problems.
- To study about NP Completeness of problems.

UNIT I ROLE OF ALGORITHMS IN COMPUTING 12

Algorithms – Algorithms as a Technology- Insertion Sort – Analyzing Algorithms – Designing Algorithms- Growth of Functions: Asymptotic Notation – Standard Notations and Common Functions- Recurrences: The Substitution Method – The Recursion-Tree Method

UNIT II HIERARCHICAL DATA STRUCTURES 12

Binary Search Trees: Basics – Querying a Binary search tree – Insertion and Deletion- Red-Black trees: Properties of Red-Black Trees – Rotations – Insertion – Deletion -B-Trees: Definition of Btrees – Basic operations on B-Trees – Deleting a key from a B-Tree- Fibonacci Heaps: structure – Mergeable-heap operations- Decreasing a key and deleting a node-Bounding the maximum degree.

UNIT III GRAPHS 12

Elementary Graph Algorithms: Representations of Graphs – Breadth-First Search – Depth-First Search – Topological Sort – Strongly Connected Components- Minimum Spanning Trees: Growing a Minimum Spanning Tree – Kruskal and Prim- Single-Source Shortest Paths: The Bellman-Ford algorithm – Single-Source Shortest paths in Directed Acyclic Graphs – Dijkstra’s Algorithm; All-Pairs Shortest Paths: Shortest Paths and Matrix Multiplication – The FloydWarshall Algorithm;

UNIT IV ALGORITHM DESIGN TECHNIQUES 12

Dynamic Programming: Matrix-Chain Multiplication – Elements of Dynamic Programming – Longest Common Subsequence- Greedy Algorithms: An Activity-Selection Problem – Elements of the Greedy Strategy- Huffman Codes.

UNIT V NP COMPLETE AND NP HARD 12

NP-Completeness: Polynomial Time – Polynomial-Time Verification – NP- Completeness and Reducability – NP-Completeness Proofs – NP-Complete Problems

TOTAL: 60 PERIODS

OUTCOMES:

Upon the completion of the course the students should be able to:

- Design data structures and algorithms to solve computing problems

- Design algorithms using graph structure and various string matching algorithms to solve real-life problems
- Apply suitable design strategy for problem solving

REFERENCES:

1. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, —Data Structures and Algorithms , Pearson Education, Reprint 2006.
2. Robert Sedgewick and Kevin Wayne, —ALGORITHMS , Fourth Edition, Pearson Education.
3. S.Sridhar, Design and Analysis of Algorithms , First Edition, Oxford University Press. 2014
4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, —Introduction to Algorithms , Third Edition, Prentice-Hall, 2011.