GUJARAT TECHNOLOGICAL UNIVERSITY
MASTER OF COMPUTER APPLICATIONS (MCA)
Semester: IV

Subject Name**: Elective I – Analysis and Design of Algorithms (ADA)**
Subject Code**: 2640010**

Learning Objectives:
- ✓ To develop a good understanding and appreciation of :
- ✓ The significance, various aspects, and fundamental issues in algorithm development.
- ✓ The need for algorithm analysis and complexity study; examples of complexity calculations; and efficiency of algorithms.
- ✓ Algorithm design using recursion and iterative methods.
- ✓ Various algorithms such as (i) Divide & Conquer, (ii) Greedy, (iii) Backtracking, (iv) Natural (evolutionary)
- ✓ Time-space trade off.
- ✓ Tractable and non-tractable problems.
- ✓ NP and NP-complete problems

Prerequisites: NIL

Contents:

1. **Introduction** **(1 Lect.)**
   Basic Concepts, Relationship between Algorithms and Other Aspects of Software, The Evolution of Algorithm

2. **Essential Mathematical Background** **(2 Lect.)**
   (A-4) Asymptotic Notations
   (A-5) Number Theory
   (A-6) Formal Languages
   (A-8) Introduction to Combinatorics

3. **Interactive Algorithm Design Issues** **(3 Lect.)**
   Introduction, Use of Loops, Efficiency of Algorithms, Estimating and Specifying Execution Time, Order Notation, Algorithm Strategies

4. **Computational Models & Design by Refinement** **(3 Lect.)**
   Introduction, Functional Model: Features of Functional Model; Recursive Processes; Analysis of Correctness and Efficiency; More Examples of Recursive Algorithms; Scope Rules; Tail Recursion and Iterative Processes; Correctness of an Iterative Process; More Examples of Iterative Processes, Imperative Model: The Primitives and the Imperative Model; Specifications and Prototyping; Examples of Step-wise Refinement

5. **Design Using Recursion** **(3 Lect.)**
   Introduction; Execution Trace: Regular Expressions; An Interesting Recursive Function Another Look at Iteration and Recursion

6. **Divide and Conquer Algorithms** **(5 Lect.)**
Introduction; A Multiplication Algorithm; Application to Graphics Algorithms: Introduction to Triangulation; Convex Hull; Where Divide & Conquer Fails; Timing Analysis

7. **Greedy Algorithms** **(4 Lect.)**
Introduction; Knapsack Problem; Job Sequencing with Deadlines; Minimum Spanning Trees: Prim's Algorithm; Kruskal's Algorithm; Union-Find Data Structures; Tree-Based Disjoint Sets and the Quick-Union Algorithm; Matroids; Dijkstra's Shortest Path Algorithm

8. **Dynamic Programming Algorithms** **(6 Lect.)**
Introduction; Multistage Graphs; Traveling Salesman; Matrix Multiplication: Brute Force Method; Dynamic Programming, Largest Common Subsequence (LCS): Brute Force Method; Dynamic Programming, Optimal Polygon Triangulation; Single Source Shortest Paths, Maximum Flow Problems: Flow Networks; Maximum Flow Problems; Analysis of Ford-Fulkerson Algorithm

9. **Backtracking Algorithms** **(4 Lect.)**
Combinatorial Search; Search and Traversal: BFS; DFS, The Backtracking Strategy: 8-Queens Problem, Backtracking Framework: Efficiency of Backtracking; M-Colouring Problem; Hamiltonian Circuits, Some Typical State Spaces: Constructing All Subsets; Constructing All Permutations; Constructing All Paths in a Graph; Bandwidth Minimization; Covering Chessboards; Convex Hull (Graham's Scan)

10. **Efficiency of Algorithms** **(4 Lect.)**
Polynomial Time (P) and Non-Polynomial Time (NPT) Algorithms; Worst and Average Case Behaviour; Time Analysis of Algorithms: Matrix Multiplication; More Timing Analysis Efficiency of Recursion; Complexity

11. **Examples of Complexity Calculations** **(4 Lect.)**
Examples from the Sorting World: Bucket Sort; Radix Sort; Simple Insertion Sort; Quick Sort; Heap Sort: Using a Tree to Sort; Merge Sort; Summary of Complexity and Characteristics of Sorting Algorithms; Complexity of Set Operations and Mappings: Set Implementation Using an Unsorted Array; Binary Search Principle; Binary Search Trees; Bit Vectors; Analysis of Hashing; The Trie Principle; Sets vs. Bags and Mappings; Amortized Analysis: Potential Functions; Examples – Binary, Binomial and Fibonacci Heaps; Binomial Heap; Fibonacci Heap; Dijsktra Shortest path Algorithm; Splay Trees: Basics of Splay Trees; Splay Operations; Amortized Timing Analysis

12. **Time-Space Trade-off** **(3 Lect.)**
Introduction; A Quick Review of Complexity; Time-Space Trade-off; Time-Space Trade-off in Algorithm Research; Case Study – Perrin Numbers: Perrin Numbers; First Try – Straight-Forward Implementation; Second Try – Dynamic Programming; Third Try – Reduction and Divide & Conquer; The Final Results

13. **Tractable and Non-tractable Problems** **(4 Lect.)**
Introduction; Upper and Lower Bounds; Efficiency and Tractability: Problem Description; A Quick and Operating Definition of NP-Complete Problems; Some Known NP-Complete Problems; What is a Certificate? Non-Deterministic Algorithms, NP-Completeness; Polynomial Time Reductions; Problem Classes: P, NP, and Others; Bounded halting Is In NPC; Cook's Theorem; Is P = NP? Approximate Solutions to NPC Problems; Provable

Intractable Problems; Even Harder Problems; Unreasonable Requirements of Memory; Complexity Classes and Intractability; Non-Computability and Un-decidability; Algorithmic Program Verification: Halting Problem (HP); Partially and Highly Un-decidable Problems; The Four Levels of Algorithmic Behaviour

14. Some NP and NP-Complete Problems (2 Lect.)

Introduction: NP-Hardness; NP-Completeness; Consequences of Being in P; Reduction Source Problems; Turing Machine: Relation between Problems and Languages; Decision Problems and Languages; Reductions: Definition of Reductions; Reduction in P; Transitivity of Reductions; NP Completeness to NP=P; Circuit Satisfiability Problem; Reduction in NPC; Steps for Proving NP Completeness; Reductions for Some Known Problems: Propositional Formulae; SAT Problem; 3-CNF SAT Problem; Cliques of a Graph; Vertex Cover (VC); Hamiltonian Cycle; Traveling Salesman Problem (TSP); Independent Sets, Certificates and Verification

Text Book:

- ✓ Parag H Dave, Himanshu B. Dave, "Design and Analysis of Algorithms", Pearson Education (2008),

Other Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", EEE, PHI (2003)
2. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, "Design and Analysis of Computer Algorithms", Pearson Education,
3. Michael T Goodrich & Roberto Tamassia, "Algorithm Design: Foundations, analysis & Internet Examples", Wiley India (2008)
4. Anany Levitin, "Introduction to Design and Analysis of Algorithms", Pearson Education,
5. Gills Brassard & Paul Bratley, "Fundamentals of Algorithms", PHI

Chapter wise Coverage from the Text Book:

Unit-1 (Chapter-1); Unit-2 (Appendices A-4, A-5, A-6, A-8); Unit-3 (Chapter-4); Unit-4 (Chapter-5); Unit-5 (Chapter-8); Unit-6 (Chapter-9); Unit-7 (Chapter-10); Unit-8 (Chapter-11); Unit-9 (Chapter-12); Unit-10 (Chapter-14); Unit-11 (Chapter-15); Unit-12 (Chapter-16); Unit-13 (Chapter-17); Unit-14 (Chapter-18).