GUJARAT TECHNOLOGICAL UNIVERSITY
(Integrated MCA)
Year V – (Semester-IX) (W.E.F. June 2017)

Subject Name: Design and Analysis of Algorithms (DAA)
Subject Code: 4490601

1. Learning Objectives:

To understand alternate methods of writing algorithms under various categories, such as Divide-and-Conquer, Dynamic Programming, Greedy Methods, Backtracking, Branch & Bound, etc. To be able to analyze algorithms by working out complexity of algorithms. To understand the basics of P, NP, and NP Complete problems

2. Prerequisites: Programming Language: C, Data Structure

3. Course Contents:

Sr. No.	Course Content	No. of Lectures
1	**Basic Concepts of Analysis and Design of Algorithms** Introduction; Overview of basic steps to solve a problem using computer; Examples; An overview of Top-down design and Recursion; Correct use of Loops in programs, Factors affecting efficiency of algorithms, Estimating and specifying execution times, Order notation: Big-oh, Omega, Theta, small omega, small-oh; **Design using Recursion**: Introduction, Execution trace through examples	**09**
2	**Algorithms Using Divide-and-Conquer Strategy** Introduction and examples; Multiplication algorithm and its analysis using various examples and data structures; Limitations of Divide-and-Conquer strategy; Timing analysis; Decrease-and-Conquer approach and examples	**07**
3	**Greedy Methods** Introduction; Examples: Knapsack problem, Job sequencing with deadlines, Minimum spanning trees, Prim's algorithm, Kruskal's algorithm, Union-find data structure, Quick-union algorithm, Shortest path, Dijkstra's shortest path algorithm, Optimal merge	**07**
4	**Dynamic Programming** Introduction; Examples: Rod cutting problem, Multistage graphs, Traveling salesman problem, Matrix multiplication, Longest common sub-sequence, Maximum flow problem	**07**
5	**Backtracking, Branch and Bound Algorithms** Combinatorial search; Search and traversal: BFS, DFS; Backtracking strategy and examples: 8-Queen problem; Backtracking framework, Efficiency of backtracking and examples; Some typical state spaces: Constructing all subsets, Constructing all permutations, Constructing all paths in a graph, Bandwidth minimization, Covering chessboards; Branch-and-Bound algorithms, Examples: Shortest path, 16-Puzzle and 8-Puzzle, Scale balancing, 0/1 Knapsack problem, Traveling salesman problem.	**10**
6	**Efficiency of Algorithms; Complexity Calculation and Categorization** Polynomial-time and Non-polynomial-time algorithms; Worst and average case behavior, Probabilistic average case analysis; Time analysis of algorithms, Examples: Matrix multiplication; Efficiency of recursion;	**09**

	Complexity: Notion of complexity, Profiling, Suppressing multiplicative constants, Counting dominant operations, Growth rate, Upper bounds, Asymptotic growth rate, The 'O' notation, Simplified definition of 'O', 'O' notation rules **Examples of Complexity Calculation**: Sorting examples: Bucket sort, Radix sort, Simple Insertion sort, Quick sort, Heap sort, Merge sort; Summary of complexity and characteristics of sorting algorithms; Complexity of set operations and mappings **Complexity Categorization of Problems**: Introduction, P, NP, NPC, NPH, etc	

4. Text Book(s):

1. Parag H Dave, Himanshu B Dave, “Design and Analysis of Algorithms”, Pearson (2014)

5. Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein, “Introduction to Algorithms”, PHI, 2nd Edition
2. Anany Levitin, “Introduction to Design and Analysis of Algorithms”, Pearson (2014)
3. S. Baase, “Computer Algorithms: Introduction to Design and Analysis”, Pearson (2002)
4. Kleinberg, “Algorithm Design”, Pearson (2013)
5. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Universities Press (2008)
6. Thomas H. Cormen, “Algorithms Unlocked”, MIT Press (2013)
7. Sanjay Dasgupta, “Algoritms”, McGraw-Hill (2006)
8. Gerard Tel, “Introduction to Distributed Algorithms”, Cambridge University Press (2004)

6. Unit wise coverage from Text book(s):

Unit 1	**Topics**
I	Chapter-2 (2.1, 2,2, 2.3.1-2.3.3), Chapter-3 (3.1, 3.5), Chapter-4 (4.2, 4.3, 4.4, 4.5), Chapter-8 (8.1, 8.2.1)
II	Chapter-9 (9.1, 9.2, 9.4, 9.5, 9.6.1, 9.6.2)
III	Chapter-10 (10.1, 10.2, 10.3, 10.4.1-10.4.8, 10.5, 10.6)
IV	Chapter-11 (11.1, 11.2, 11.3, 11.4, 11.5, 11.6, 11.9)
V	Chapter-12 (12.1, 12.2, 12.3, 12.4, 12.5.1-12.5.5, 12.6.1, 12.6.3, 12.6.4, 12.6.6, 12.6.7)
VI	Chapter-14 (14.1, 14.2, 14.3, 14.3.1, 14.4, 14.5, 14.5.1-14.5.11); Chapter-15 (15.1, 15.1.1-15.1.6, 15.2, 15.3); Chapter-17 (17.1)

7. Accomplishment of the student after completing the course:

The student will be able to decide on an appropriate category of algorithms for solving a given problem. With an understanding of more than one method of solving problems using algorithms, (s)he will be able to carry out complexity of algorithms and decide on an efficient algorithm for the task on hand. (S)he will also have an idea about categorization of problems into P, NP, NPC, NPH.