CS22403 DESIGN AND ANALYSIS OF ALGORITHMS

L	T	P	C
3	0	0	3

OBJECTIVES

- Analyze the asymptotic performance of various algorithms
- Design algorithms using divide and conquer approach
- Design algorithms using dynamic programming and greedy technique
- Understand the methods for iterative improvement and computational geometry
- Design algorithms using Backtracking and Brach and bound approach and learn about Class P, NP, NP-complete and NP-hard.

UNIT I INTRODUCTION 9

The Role of Algorithms in Computing: Algorithms as a technology - Fundamentals of algorithmic problem solving – Analysis framework – Growth of functions: Asymptotic notation – Standard notations and common functions. Mathematical analysis of Non recursive and recursive algorithms- Insertion Sort algorithm and analysis – Brute force approach: Sequential search, Traveling Salesman Problem, Knapsack problem.

UNIT II DIVIDE AND CONQUER 9

The Substitution Method for Solving Recurrences – The Recursion-Tree method for Solving Recurrences- The Master Method for Solving Recurrences – Merge sort, Quick sort algorithm and analysis – The maximum-sub array problem – Finding Closest Pair of Points.

UNIT III DYNAMIC PROGRAMMING, GREEDY TECHNIQUE AND STRING MATCHING ALGORITHMS 9

Dynamic Programming: Knapsack Problem and memory functions, Longest common subsequence, Optimal Binary Search Tree, Warshall’s and Floyd’s Algorithm. Greedy Technique: Minimum Spanning Trees – Kruskal’s and Prim’s Algorithms, Single source Shortest Paths - Dijkstra's Algorithm, Huffman Trees. String Matching algorithms: The naïve approach, Rabin-Karp algorithm, Knuth- Morris-Pratt algorithm.

UNIT IV ITERATIVE IMPROVEMENT AND COMPUTATIONAL GEOMETRY 9

Maximum Flow: Flow networks, Ford Fulkerson method, Maximum Bipartite matching – Linear Programming: Standard and Slack Forms, The Simplex Algorithm, Geometric Interpretation of Linear Programming - Computational Geometry: Line Segment Properties, Graham Scan, Jarvis’s March.

UNIT V COPING WITH LIMITATIONS OF ALGORITHMIC POWER AND NP-COMPLETENESS 9

Backtracking: N-Queen’s problem, Subset-sum problem – Branch and bound: Knapsack problem, Traveling salesman problem - NP-completeness and the classes P and NP: Polynomial Time, Polynomial Time

Verification, Circuit Satisfiability, Formula satisfiability, 3-CNF satisfiability, The Clique Problem, Vertex Cover.

TOTAL (L:45): 45 PERIODS

OUTCOMES:

CO	CO statements Upon successful completion of the course, the students should be able to	RBT Level
CO1	Analyze the running time of algorithms using asymptotic analysis.	**4**
CO2	Apply the divide-and-conquer techniques and analyze the running time of the algorithms in real-world problems.	**3**
CO3	Apply the dynamic programming and greedy paradigms and analyze the running time of the algorithms using those techniques.	**3**
CO4	Employ iterative improvement and computational geometry methods to solve engineering problems.	**3**
CO5	Describe the limitations of algorithm power and methods to cope with the limitations of algorithm power for various problems	**2**

1- Remember, 2- Understand, 3- Apply, 4- Analyse, 5- Evaluate, 6- Create

TEXT BOOKS

1. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to Algorithms”, Third Edition, PHI Learning Private Limited, 2012.
2. Anany Levitin, “Introduction to the Design and Analysis of Algorithms”, Third Edition, Pearson Education, 2012.

REFERCENCES

1. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education, Reprint 2006.
2. Donald E. Knuth, “The Art of Computer Programming”, Volumes 1& 3 Pearson Education, 2009.
3. Steven S. Skiena, “The Algorithm Design Manual”, Second Edition, Springer, 2008.
4. http://nptel.ac.in/