III Year I Semester

Code: 17CS502

L	T	P	C
3	1	0	3

DESIGN OF ALOGORITHMS AND ANALYSIS

OBJECTIVES:

Upon completion of this course, students will be able to do the following:

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Apply important algorithmic design paradigms and methods of analysis.
5. Synthesize efficient algorithms in common engineering design situations

UNIT-I:

Introduction: Examples and motivation, Asymptotic complexity: informal concepts, formal notation, examples

Searching and Sorting: binary search, insertion sort, selection sort, merge sort, quick sort, stability and other issues.

UNIT-II:

Graphs: Motivation, Directed acyclic graphs, Graph exploration: BFS, DFS, applications.

UNIT-III:

Search Trees: Introduction, Traversals, insertions, deletions, Balancing, Priority queues, heaps

UNIT-IV:

Greedy : Interval scheduling, Minimum cost spanning trees: Prim's algorithm, Kruskal's Algorithm, Shortest paths: un weighted and weighted, Single source shortest paths: Dijkstra, Huffman coding.

UNIT-V:

Dynamic Programming: weighted interval scheduling, memorization, 0/1 knapsack,Travelling Salesman Problem, matrix chain multiplication, shortest paths: Bellman Ford, Floyd, Warshall.

UNIT-VI:

Backtracking: The General Method, The 8-Queens Problem, Sum of Subsets, Graph Coloring , Hamiltonian Cycles.

OUTCOMES:

Students who complete the course will have demonstrated the ability to do the following:

- Argue the correctness of algorithms using inductive proofs and invariants.
- Analyze worst-case running times of algorithms using asymptotic analysis.
- Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and

conquer algorithms. Derive and solve recurrences describing the performance of divide and-conquer algorithms.

- Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic programming algorithms, and analyze them.

TEXT BOOKS:

1. Fundamentals of computer algorithms E. Horowitz S. Sahni, University Press
2. Introduction to AlgorithmsThomas H. Cormen, PHI Learning

REFERENCE BOOKS

1. The Design and Analysis of Computer Algorithms, Alfred V. Aho, John E. Hopcroft, Jeffrey D.Ullman
2. Algorithm Design, Jon Kleinberg, Pearson.