Algorithms
Core Course for M.Tech
Number of credits: 4

Objectives: Appreciate inherent hardness of a problem and learn strategies available for problem solving; design algorithms and give proof of correctness, implementation of the algorithms by choosing appropriate data structures; gaining skills to analyze the complexity of an algorithm by giving mathematical proofs where necessary.

At the end of the course, students should be able to appreciate the inherent structure/hardness of a problem and choose an appropriate strategy to solve the problem and design an algorithm that suits the time coplexity requirements of the problem.

Pre-requisites: Data structures, Discrete mathematical structures, knowledge of sorting algorithms and basic search strategies

Syllabus

- Analysis of Algorithms:

  Asymptotic Notation; Best, worst and average case analysis of algorithms; Solving recurrence relations using substituion method, generating functions, Master's theorem etc.

  Warm-up to complexity analysis: Heap data structure, priority queue application, *Best, worst and average case* analysis of a few sorting algorithms like heap sort, insertion, bubble, selection, counting and radix sort algorithms

- Strategies for problem solving:

  a) Divide and Conquer strategy: Time complexity analyis for Merge Sort and Quick Sort Algorithms,

  b) Greedy strategy: Theoretical foundation of greedy strategy: Matroids Algorithms for solving problems like Knapsack Problem

(Fractional), Minimum Spanning Tree problem; Shortest Paths, Job Scheduling, Huffman's code etc along with proofs of corrections and complexity analysis

c) Dynamic Programming strategy: Identify situations in which greedy and divide and conquer strategies may not work. Understanding of optimality principle. Technique of memoization. Applications to problems like Coinchange, 0/1 and 0/n- Knapsack, Shortest Paths, Optimal Binary Search Tree(OBST), Chained Matrix Multiplication, Traveling Salesperson Problem(TSP) etc.

d) Backtracking and Branch & Bound strategies: State space tree construction, traversal techniques and solving problems like 0/1 and 0/n knapsack, TSP, Applications of Depth First Search: Topological sorting, Finding strongly connected components and game problems.

- Theory of NP-Completeness: Complexity classes of P, NP, NP-Hard, NP-Complete, Polynomial reductions, Cook's theorem. Discussion of problems: Sastisfiability(SAT), CNF-SAT, Min-Vertex Cover, Max-Clique, Graph Coloring, NP-Completeness proofs.

Text books:

Introduction to Algorithms-T.Cormen, C.E.Leiserson, R.L.Rivest,PHI, 3rd Edition 2009.
Algorithms- R.Johnsonbaugh and M.Schaefer, Pearson, 2004.
Fundamentals of Algorithmics - G.Brassard and P.Bratley, PH, 1996

Reference books:

The Algorithm Design Manual- Steven S. Skiena, Springer, 2009