UE20CS251: Design and Analysis of Algorithms (4-0-0-4-4)

Algorithms play a key role in science and practice of computing. Learning algorithm design technique is a valuable endeavour from practical standpoint and algorithm design techniques have considerable utility as general problem solving strategies, applicable to problems beyond computing. This course includes classic problems of computer science, application of design techniques and analysis in terms of time and space.

Course Objectives:

- Learn to design and analyze algorithms with an emphasis on the resource utilization in terms of time and space.
- Learn various techniques in the development of algorithms so that the effect of problem size and architecture design on the efficiency of the algorithm is appreciated.
- Learn to apply appropriate algorithmic design techniques for specific problems.
- Learn to trade space for time in algorithmic design using input enhancement and per-structuring.
- Learn to improve the limitations of algorithmic power.

Course Outcomes:

At the end of the course, the student will be able to:

- Identify the design technique used in an algorithm.
- Analyze algorithms using quantitative evaluation.
- Design and implement efficient algorithms for practical and unseen problems.
- Analyze time efficiency over trading space.
- Understand the limits of algorithms and the ways to cope with the limitations.

Course Content:

Unit 1 : Introduction

Algorithms, Fundamentals of Algorithmic Problem Solving, Important Problem Types. Analysis of Algorithm Efficiency: Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non - Recursive and Recursive Algorithms. -**12 Hours**

Unit 2 : Brute Force and Divide-and-Conquer

Brute Force: Selection Sort, Bubble Sort, Sequential Search, Brute Force String Matching, Exhaustive Search. Divide-and-Conquer: Master Theorem, Merge Sort, Quick Sort, Binary Search, Binary Tree Traversals, Complexity analysis for finding the height of BST, Multiplication of Large Integers, Strassen's Matrix Multiplication.- **12 Hours**

Unit 3 : Decrease-and-Conquer & Transform-and-Conquer

Decrease-and-Conquer: Insertion Sort, Topological Sorting, Algorithms for Generating Combinatorial Objects, Decrease-by-a-Constant-Factor Algorithms. Transform-and-Conquer: Pre-sorting, Heap Sort, Red-Black Trees, 2-3 Trees and B Trees. Problems on - Decrease by a constant factor /constant number.**- 10 Hours**

Unit 4 : Space and Time Tradeoffs & Greedy Technique

Space and Time Tradeoffs: Sorting by Counting, Input Enhancement in String Matching - Horspool's and Boyer-Moore Algorithms. Greedy Technique: Prim's Algorithm, Kruskal's Algorithm and union-find algorithm, Dijkstra's Algorithm, Huffman Trees. **10 Hours**

Unit 5 : Limitations & Coping with the Limitations of Algorithm Power
Limitations of Algorithm Power: Lower-Bound Arguments, Decision Trees, P, NP, and NP-Complete, NP-Hard Problems. Coping with the Limitations of Algorithm Power: Backtracking, Branch-and-Bound. Dynamic Programming: Computing a Binomial Coefficient, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms.-**12 Hours**

Tools / Languages: C Language, GCC Compiler.

Text Book:
1: "Introduction to the Design and Analysis of Algorithms", Anany Levitin, Pearson Education, Delhi (Indian Version), 3rd edition, 2012.

Reference Book(s):
1: "Introduction to Algorithms", Thomas H Cormen, Charles E Leiserson, Ronald L Rivest and Clifford Stein, Prentice-Hall India, 3rd Edition, 2009.
2: "Fundamentals of Computer Algorithms", Horowitz, Sahni, Rajasekaran, Universities Press, 2nd Edition, 2007.
3: "Algorithm Design", Jon Kleinberg, Eva Tardos, Pearson Education, 1st Edition, 2006.