# 19CS2202 - DESIGN AND ANALYSIS OF ALGORITHMS

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Category:** | Program Core | **Credits:** | 3 |
| **Course Type:** | Theory | **Lecture - Tutorial - Practical:** | 3-0-0 |
| **Prerequisite:** | Knowledge on concept of preparing algorithms for basic problems, elementary data structures and their associated operations. | **Sessional Evaluation:**  **Univ. Exam Evaluation:**  **Total Marks:** | 40  60  100 |
| **Objectives** | * To understand the design and performance issues of an algorithm. * To be familiar with the kinds of design techniques. * To compare the design methods for producing optimal solution for real world problems. * To understand the various computational models for an effective design. * To learn to design the solutions for NP hard and NP complete problems. | | |

|  |  |  |
| --- | --- | --- |
| **Course Outcomes** | Upon successful completion of the course, the students will be able to: | |
| CO1 | Analyze the time and space complexity of an algorithms. |
| CO2 | Design and analysis of algorithms using greedy strategy. |
| CO3 | Identify dynamic programming design methodology to solve problems involving principle of optimality. |
| CO4 | Perform operations on sets and tree structures and also to understand their applications. |
| CO5 | Solve problems by constructing a state space tree with branch and bound and backtracking. |
| CO6 | Analyze the classes P, N and NP Complete and be able to prove that a certain problem is NP complete. |
| **Course Content** | UNIT-I  **Introduction:** Algorithm analysis, Performance analysis, Space complexity and time complexity, Big „O‟ notation, Omega notation, Theta notation, Different mathematical approach‟s for solving Time complexity of Algorithms.  **Divide and conquer:** General method, Binary search, Merge sort, Quick sort, Strassen‟s matrix multiplication.  UNIT-II  **Greedy method:** General method, Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees, Optimal storage on tapes, Single source shortest paths.  UNIT-III  **Dynamic programming:** General method, Multistage graphs, All pairs shortest paths, Single source shortest paths. 0/1 Knapsack problem, Reliability design problem, Travelling sales person problem.  UNIT-IV  **Basic Traversal & Search Techniques:** Techniques for Binary Trees and Graphs, Connected Components and Spanning Tress, Bi-Connected Components and DFS.  **Sets and Disjoint set Union:** Introduction, Union and Find operations.  UNIT-V  **Back tracking:** General method, N-Queens problem, Sum of subsets, Graph coloring problem.  **Branch and bound:** General method, Least cost (LC) search, Control abstractions for  LC search, 0/1 Knapsack problem, Travelling salesperson problem.  UNIT-VI  **NP Hard and NP complete problems:** Basic concepts - Nondeterministic algorithms, The classes NP hard and NP complete; NP hard graph problems - Clique decision problem, Node cover decision problem.  **PRAM Algorithms:** Introduction, Computational Model. | |
| **Text Books and References:** | Text Books:   1. E.Horowitz, S.Sahni, S.Rajasekaran, ”Fundamentals of Computer Algorithms”, 2ndEdition, Universities Press, ISBN: 978-8173716126, 2008. | |
| Reference Books:   1. Thomas H.Cormen, Charles E.Leiserson, Ronald L.Rivest, Cliford Stein, 2. “Introduction to Algorithms”, 3rd Edition, Prentice-Hall of India, ISBN: 978-81- 203-4007-7, 2010 3. S.Sridhar, “Design and Analysis of Algorithms”, Oxford University Press, India, ISBN - 13: 978-0-19-809369-5, ISBN-10: 0-19-809369-1, 2015 | |
| **E-Resources** | 1. <https://nptel.ac.in/courses> 2. <https://freevideolectures.com/university/iitm> | |