Details of Course:

Course Title	Course Structure			Pre-Requisite
MC 206: Algorithm Design and Analysis	L	T	P	Data Structures
	3	1	0	

Course Objective: To introduce the concept of algorithmic efficiency by analyzing various algorithms such as Searching,Sorting, Divide-and-Conquer algorithms and to know detail about Greedy Paradigm, Principle of Dynamic Programming, Back Tracking, Branch and Bound, and Computational Complexity.

Course Outcome (CO):

CO1	Design efficient algorithms for real-life problems using different algorithmic paradigms and identify the limitations of each algorithmic paradigms for problem solving
CO2	Design and analyze the running time of algorithms in terms of asymptotic notation.
CO3	Describe different paradigms of algorithm design, such as Divide & Conquer, Greedy, Dynamic Programming, etc., and conclude the correctness.
CO4	Compare the notion of tractable and intractable problems and develop algorithms for computationally intractable problems.
CO5	Solve and analyze the inter-disciplinary real-world problems including sorting problems, trees and graphs problems, and recurrence relations.

S. No.	Contents	Contact hours
1.	Introduction: Concept of algorithmic efficiency, run time analysis of algorithms, Asymptotic Notations. Growth of Functions, Recurrence Relation, Master's Theorem, Correctness of Algorithm. Divide and Conquer Approach: Introduction, Analysis of Run time and Correctness of divide and conquer based Searching and Sorting algorithms, Heap sort, Strassen's matrix multiplication.	8
2.	Greedy Method: Overview of the greedy paradigm examples of exact optimization solution: minimum cost spanning tree, approximate solutions: Knapsack problem, Kruskal's algorithm and Prim's algorithm for finding Minimum cost spanning tree, Dijkstra's algorithm for single source shortest path problem.	9
3.	Dynamic programming: Principle of dynamic programming. Applications: Bellman Ford Algorithm for single source shortest path problem, Floyd-Warshall algorithm for all pair shortest path problem, 0/1 Knapsack Problem, Matrix chain multiplication, Traveling salesman Problem, longest Common sequence (LCS).	8
4.	Back tracking: Overview, 8-queen problem, and 0/1 Knapsack problem, Subset Sum Problem, Traveling Salesman problem. Branch and bound: LC searching Bounding, FIFO branch and bound, LC branch and bound application: 0/1 Knapsack problem.	8

5.	Computational Complexity: Complexity measures, Polynomial Vs non-polynomial time complexity; NP-hard and NP-complete classes, examples: Circuit Satisfiablity, Vertex cover, Subset Sum problem, Randomized Algorithms, String Matching, NP-Hard and NP-Completeness, Approximation Algorithms, Sorting Network, Matrix Operations, Polynomials and FFT, Number Theoretic Algorithms.	9
	Total	42

Suggested Books:

S. No.	Name of Books/Authors/Publishers	Year of Publication
1.	Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. "Introduction to Algorithms", MIT Press.	4th edition 2022
2.	Horowitz, Ellis, Sartaj Sahni, and Sanguthevar Rajasekaran. "Computer algorithms C++: C++ and pseudocode versions" Macmillan.	1997
3.	Sara Baase and Allen Van Gelder. "Computer algorithms: introduction to design and analysis" Pearson Education India, 2009.	3rd edition 2009