CS205: Design and Analysis of Algorithms	**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.

S. No	Course Outcomes (CO)
CO1	To evaluate time and space complexity of recursive and non-recursive algorithms.
CO2	To analyze various divide and conquer algorithms and construct recurrence relations.
CO3	To design and analyze greedy algorithm to solve real life problems.
CO4	To analyze different algorithms to find minimum spanning tree and shortest path algorithm.
CO5	To apply dynamic programming techniques to solve numerous optimization problems
CO6	To design backtracking and branch and bound techniques for solving specific problems.
CO7	To be able to classify problems as P, NP, NP complete and compose approximate.

S. No	Contents	Contact Hours
UNIT 1	Introduction: Concept of algorithmic efficiency, run time analysis of algorithms, Asymptotic Notations. Growth of Functions, Master's Theorem	**6**
UNIT 2	Searching and Sorting: Structure of divide-and-conquer algorithms; examples: binary search, quick sort, Strassen Matrix Multiplication; merge sort, heap sort and Analysis of divide and conquer run time recurrence relations, Application of graph theory concepts- connected components, Cut vertex, Bridge	**8**
UNIT 3	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 Trees, Dijkstra's and Bellman Ford Algorithm for finding Single source shortest paths, Huffman coding, Activity Selection Problem.	**10**
UNIT 4	Dynamic programming: Principles of dynamic programming. Applications: Rod cutting problem, Floyd-Warshall algorithm for all pair shortest paths. Matrix multiplication, Travelling salesman Problem, Longest Common sequence, Back tracking: Overview, 8-queen problem, and Knapsack problem, Edit Distance Problem, Rod cutting problem.	**10**
UNIT 5	Branch and bound: LC searching Bounding, FIFO branch and bound, LC branch and bound application: 0/1 Knapsack problem.	**8**
UNIT 6	Computational Complexity: Complexity measures, Polynomial Vs non-polynomial time complexity; NP-hard and NP-complete classes, examples: Circuit Satisfiability, Vertex cover, Subset Sum problem	**6**
	Total	**48**