CP302: DESIGN AND ANALYSIS OF ALGORITHMS
CREDITS = 6 (L=4, T=0, P=2)

Course Objective:

To impart the knowledge of different algorithm design techniques and to analyze complexity of given algorithms.

Teaching and Assessment Scheme:

Teaching Scheme			Credits	Assessment Scheme				Total Marks
				Theory		Practical		
L	T	P	C	ESE	CE	ESE	CE	
4	0	2	6	70	30	30	20	150

Course Content:

Unit No.	Topics	Teaching Hours
1	**Analysis of Algorithms and Mathematics:** Introduction, Mathematics for Algorithmic Sets, Functions and Relations, Vectors and Matrices, Linear Inequalities and Linear Equations, Average, Best and worst case analysis, Amortized analysis , Asymptotic Notations, Analyzing control statement, Loop invariant and the correctness of algorithms, Sorting Algorithms and analysis: Bubble sort, Selection sort, Insertion sort, Shell sort, Heap sort, Sorting in linear time : Bucket sort, Radix sort and Counting sort, other related data structure like binomial heap, Fibonacci heap, read-black tree.	12
2	**Divide and Conquer Algorithms:** Introduction, Recurrence and different methods to solve recurrence, Multiplying large Integers Problem, Problem Solving using divide and conquer algorithm - Binary Search, Max-Min problem, Sorting (Merge Sort, Quick Sort), Matrix Multiplication, Exponentiation.	06
3	**Dynamic Programming:** Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming – Calculating the Binomial Coefficient, Making Change Problem, Assembly Line-Scheduling, Knapsack problem, All Points Shortest path, Matrix chain multiplication, Longest Common Subsequence.	07

4	**Greedy Algorithms & Graph:** General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm, Activity selection problem, Elements of Greedy Strategy, The Knapsack Problem, Job Scheduling Problem, Huffman code, Undirected Graph, Directed Graph, Traversing Graphs, Depth First Search, Breath First Search, Topological sort, Connected components, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Shortest path finding algorithms.	13
5	**Backtracking, Branch and Bound:** Introduction, The Eight queens problem, Knapsack problem, Travelling Salesman problem, Minimax principle.	03
6	**String Matching:** Introduction, The naive string matching algorithm, The Rabin-Karp algorithm, String Matching with finite automata, The Knuth-Morris-Pratt algorithm.	04
7	**Introduction to NP-Completeness:** The class P and NP, Polynomial reduction, NP- Completeness Problem, NP-Hard Problems. Travelling Salesman problem, Hamiltonian problem, Approximation algorithms. Solving NP problems.	08
8	**Linear Programming:** Introduction, Duality of Problem, Simplex Method, Big M Method, Introduction to Integer Programming, Diet Problem, Transportation Problem, Activity Selection Problem.	07
	TOTAL	**60**

List of References:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "*Introduction to Algorithms*", PHI.
2. Gills Brassard, Paul Bratley, "*Fundamental of Algorithms*", PHI.
3. Anany Levitin, "*Introduction to Design and Analysis of Algorithms*", Pearson.
4. Shailesh R Sathe ,"*Foundations of Algorithms*", Penram publications
5. Dave and Dave, "*Design and Analysis of Algorithms*", Pearson publication.

Course Outcome (COs):

After learning the course the students will be able to:

1. Analyze the asymptotic performance of algorithms.
2. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
3. Find optimal solution by applying dynamic programming & greedy approaches
4. Apply pattern matching algorithms to find particular pattern from given data
5. Differentiate polynomial and non-polynomial problems.
6. Apply graphs to model engineering problems & design appropriate algorithm to solve those problems.