Name of Department:- Computer Science and Engineering

1. Subject Code: TCS 409 Course Title: **Design and Analysis of Algorithms**
2. Contact Hours: L: 3 T: - P: -
3. Semester: VI

4. Pre-requisite: TCS 101, TCS 201,TCS 302

5. Course Outcomes: After completion of the course students will be able to

1. Understand various asymptotic notations to analyze time and space complexity of algorithms
2. Analyze the various paradigms for designing efficient algorithms using concepts of design and conquer, greedy and dynamic programming techniques
3. Provide solutions to complex problems using the concept of back tracking and branch and bound techniques.
4. Apply algorithm design techniques to predict the complexity of certain NP complete problems.
5. Implement Dijkstra's, Bellman-ford, Prims, Kruskal's algorithms to solve the real world problems like traveling salesman problem, job sequencing, packet routing etc
6. Apply pattern matching algorithms like Rabin Karp Algorithm, Brute-force techniques etc to find a particular pattern.

Detailed Syllabus

UNIT	CONTENTS	Contact Hrs
Unit – I	**Asymptotic Notations and Searching Algorithms** **Introduction to Algorithms -** What is an Algorithm, Rate of growth, Commonly used rate of growths, Types of analysis, Asymptotic Notations, Master theorem **Searching -** Linear search (sorted and unsorted), Iterative and recursive binary search, Tower of Hanoi and solving its recursion, Fibonacci and solving its recursion	**8**
Unit - II	**Sorting Algorithms Sorting -** Bubble sort, Insertion sort, selection sort, quick sort, randomized quick sort, merge sort, heap sort, counting sort, External sorting **Divide sorting algorithms into following types -** online sort, stable sort, in place sort, Comparison of sorting algorithms on the basis of number of swaps, by number of comparisons, recursive or iterative nature, time and space complexity	**10**
Unit – III	**Graph Algorithms**	**12**

	Representation of Graphs, Breadth-first search (BFS), depth-first search (DFS), topological sort, Difference between BFS and DFS **Data structures for disjoint sets -** Finding cycle in a graph, Finding strongly connected components **Minimum spanning trees -** Kruskal and Prim algorithms (Greedy Algorithms) **Single source shortest paths -** Dijkstra (Greedy Approach) and Bellman ford (Dynamic Programming) algorithms **All pair shortest paths -** The Floyd Warshall algorithm	
Unit – IV	**Algorithm Design Techniques - Greedy and Dynamic Programming** **Greedy algorithms -** Activity selection problem, Job sequencing problem, Huffman codes, fractional knapsack problem **Dynamic Programming -** Overlapping substructure property, Optimal substructure property, Tabulation vs Memoization, Fibonacci numbers, 0/1 Knapsack problem, Longest common subsequence, Matrix chain multiplication	**10**
Unit – V	**Hashing, String Matching and NP-Completeness** **Hashing Data Structure -** Introduction to Hashing, Hash function, Collision and collision handling, Collision handling - Chaining, Open addressing **String Matching -** Naive string-matching algorithm, The Rabin-Karp algorithm, The Knuth-Morris-Pratt algorithm **NP-Completeness** - Importance of NP-completeness, P, NP, NP Complete and NP hard problems, Polynomial time and polynomial time verification, The subset-sum problem, The traveling salesman problem	**10**
	Total	**50**

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein:" Introduction to Algorithms", 2nd Edition, PHI, 2006.

Reference Books:

1. Donald E.Knuth:"The Art of Computer Programming: Volume 1: Fundamental Algorithms",3rd Edition
2. Ellis Horowitz, Sartaj Sahni, SanguthevarRajasekaran:" Fundamentals of Computer Algorithms", 2nd Edition, University press, 2007.
3. Anany Levitin: "Introduction to the Design & Analysis of Algorithms", 2nd Edition, Pearson Education, 2007.