CE355: DESIGN & ANALYSIS OF ALGORITHMS

Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	4	2	-	6	**5**
Marks	100	50	-	150	

Pre-requisite courses:

- Data Structure and Algorithms
- Programming language

Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Algorithm Analysis	10
2.	Greedy Algorithm	08
3.	Dynamic Programming	10
4.	Divide and Conquer Algorithm	07
5.	Exploring Graphs	10
6.	String Matching and Introduction to NP-Completeness	08
7.	Approximation Algorithms	07
	Total hours (Theory) :	**60**
	Total hours (Lab) :	30
	Total hours :	90

Detailed Syllabus:

1.	**Introduction**	**10 Hours**	**17%**
	Fundamentals of algorithms, Performance Analysis, Primitive Operations, Time Complexity and Space Complexity, The efficiency of algorithm, average and worst case analysis, elementary operation, Asymptotic Notation, Analysing control statement, Analysing Algorithm using Barometer, Solving		

	recurrence Equation, Sorting Algorithm: Selection, Insertion, Bubble Sort		
2.	**Greedy Algorithm**	**08 Hours**	**13%**
	Greedy: Characteristics, greedy functions, Problem solving: Making change problem, The Knapsack Problem, Dijkstra's Shortest paths; Job Scheduling Problem, Disjoint sets, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm, Huffman coding		
3.	**Dynamic Programming**	**10 Hours**	**17%**
	Dynamic Programming: The Principle of Optimality, Problem Solving: Calculating the Binomial Coefficient, Making Change Problem, Assembly Line-Scheduling Knapsack Problem, Shortest Path Matrix Chain Multiplication, Longest Common Subsequence, All Pairs Shortest Path (Floyd-Warshall), Travelling Salesman Problem, Bellman Ford Algorithm.		
4.	**Divide and Conquer Algorithm**	**07 Hours**	**12%**
	Multiplying large Integers Problem, Binary Search Sorting (Merge Sort, Quick Sort), Matrix Multiplication, Exponential		
5.	**Exploring Graphs**	**10 Hours**	**17%**
	An Introduction, Undirected Graph, Directed Graph, Breath First Search, Depth First Search, Graph coloring problem, Applications of BFS & DFS, Backtracking –The Knapsack Problem; The Eight Queens problem, Branch and Bound –The Assignment Problem, The Knapsack Problem		
6.	**String Matching and Introduction to NP-Completeness**	**08 Hours**	**13%**
	The naïve string-matching algorithm, The Rabin-Karp algorithm, KMP Algorithm for Pattern Searching, Boyer–Moore string-search algorithm**,** The class P and NP Problem, Polynomial reduction, NP-Completeness Problem, NP-Hard problems		
7.	**Approximation Algorithms**	**07 Hours**	**11%**
	Vertex Cover Problem, Travelling Salesman Problem, Set Covering Problem, Randomization and Linear Programming		

Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Analyse the asymptotic performance of algorithms.
CO2	Derive time and space complexity of different sorting algorithms and compare them to choose application specific efficient algorithm.
CO3	Understand and analyse the problem to apply design technique from divide and conquer, dynamic programming, backtracking, branch and bound techniques and understand how the choice of algorithm design methods impact the performance of programs.
CO4	Understand and apply various graph algorithms for finding shorted path and minimum spanning tree.
CO5	Synthesize efficient algorithms in common engineering design situations.
CO6	Understand the notations of P, NP, NP-Complete and NP-Hard.

Sr. No	**Course Outcomes (Cos)**	**Employability/ Entrepreneurship/ Skill development**
1.	Analyse the asymptotic performance of algorithms.	Skill Development
2.	Derive time and space complexity of different sorting algorithms and compare them to choose application specific efficient algorithm.	Skill Development
3.	Understand and analyse the problem to apply design technique from divide and conquer, dynamic programming, backtracking, branch and bound techniques and understand how the choice of algorithm design methods impact the performance of programs.	Skill Development

4.	Understand and apply various graph algorithms for finding shorted path and minimum spanning tree.	Skill Development
5.	Synthesize efficient algorithms in common engineering design situations.	Employability
6.	Understand the notations of P, NP, NP-Complete and NP-Hard.	Skill Development

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	-	-	-	-	-	-	-	-	-	-	1	-
CO2	2	2	-	-	-	-	-	-	-	-	-	2	2	-
CO3	3	3	3	3	2	-	-	-	-	-	-	2	2	-
CO4	2	3	3	1	-	-	-	-	-	-	-	-	2	-
CO5	1	-	1	-	-	-	-	-	-	-	-	2	1	1
CO6	3	1	-	-	-	-	-	-	-	-	-	-	1	-

Enter correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

If there is no correlation, put "-"

Recommended Study Material:

- **Text books:**
  1. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald Rivest and Clifford Stein, MIT Press

- **Reference books:**
  1. Fundamental of Algorithms by Gills Brassard, Paul Bratley, Pentice Hall of India.
  2. Fundamental of Computer Algorithms by Ellis Horowitz, Sartazsahni and

sanguthevar Rajasekarm, Computer Sci.P.

3. Design & Analysis of Algorithms by P H Dave & H B Dave, Pearson Education.

❖ **Web materials:**

1. http://highered.mcgraw-hill.com/sites/0073523402/

❖ **Software:**

1. Code::Blocks / Online C Editor