CE342: DESIGN & ANALYSIS OFALGORITHMS

Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	2	-	5	4
Marks	100	50	-	150	

A. Pre-requisite courses:

- Data Structure and Algorithm Design
- Computer Programming

B. Outline of the course:

Sr. No.	Title of the unit	Minimum number of hours
1.	To derive time and space complexity of algorithm.	03
2.	Analysis of Algorithm	06
3.	Greedy Algorithm	07
4.	Divide and Conquer Algorithm	07
5.	Dynamic Programming	08
6.	Exploring Graphs	04
7.	Backtracking & Branch & Bound	05
8.	String Matching and Introduction to NP- Completeness	05

Total hours (Theory): 45
Total hours (Lab): 30
Total hours : 75

C. Detailed Syllabus:

1.	**Basics of Algorithms and Mathematics**	**03 Hours**	**05 %**
1.1	What is an algorithm?		
1.2	Performance Analysis, Model for Analysis - Random Access Machine (RAM), Primitive Operations		
1.3	Time Complexity and Space Complexity		
2.	**Analysis of Algorithm**	**06 Hours**	**14 %**
2.1	The efficiency of algorithm, average and worst case analysis, elementary operation		
2.2	Asymptotic Notation		
2.3	Analyzing control statement		
2.4	Analyzing Algorithm using Barometer		
2.5	Solving recurrence Equation		
2.6	Sorting Algorithm		
3.	**Greedy Algorithm**	**07 Hours**	**16 %**
3.1	General Characteristics of greedy algorithms		
3.2	Problem solving using Greedy algorithm		
3.3	Making change problem		
3.4	Graphs: Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm		
3.5	Graphs: Shortest paths; The Knapsack Problem; Job Scheduling Problem		
4.	**Divide and Conquer Algorithm**	**07 Hours**	**16 %**
4.1	Multiplying large Integers Problem		
4.2	Binary Search		
4.3	Sorting (Merge Sort, Quick Sort)		
4.4	Matrix Multiplication		
4.5	Exponential		

5.	**Dynamic Programming**	**08 Hours**	**18 %**
5.1	Introduction, The Principle of Optimality		
5.2	Problem Solving using Dynamic Programming – Calculating the Binomial Coefficient		
5.3	Making Change Problem		
5.4	Assembly Line-Scheduling		
5.4	Knapsack Problem		
5.5	Shortest Path		
5.6	Matrix Chain Multiplication		
5.7	Longest Common Subsequence		
6.	**Exploring Graphs & Backtracking**	**04 Hours**	**09 %**
6.1	An introduction using graphs and games,		
6.2	Traversing Trees – Preconditioning Depth First Search-Undirected Graph; Directed Graph, Breath First Search, Applications of BFS &DFS		
7.	**Backtracking & Branch & Bound**	**05 Hours**	**12%**
7.1	Backtracking –The Knapsack Problem; The Eight queens problem, General Template		
7.2	Brach and Bound –The Assignment Problem; The Knapsack Problem, The min-max principle		
8.	**String Matching and Introduction to NP-Completeness**	**05 Hours**	**10%**
8.1	The naïve string matching algorithm		
8.2	The Rabin-Karp algorithm		
8.3	The class P and NP Problems		
8.4	Polynomial reduction		
8.5	NP- Completeness Problem		
8.6	NP-Hard problems		

D. Instructional Method and Pedagogy:

- At the start of course, the course delivery pattern, prerequisite of the subject will be discussed.
- Lectures will be conducted with the aid of multi-media projector, black board, OHP etc.

- It deals with guided discovery problems through intriguing puzzles to solve, structured hands on activities, carefully worded leading questions, crucial hints, guided discovery problems escort students step by step through process of scientific discovery.
- Attendance is compulsory in lectures and laboratory which carries a 5% component of the overall evaluation.
- Minimum two internal exams will be conducted and average of two will be considered as a part of 15% overall evaluation.
- Assignments based on course content will be given to the students at the end of each unit/topic and will be evaluated at regular interval. It carries a weight-age of5%.
- Surprise tests/Quizzes will be conducted which carries 5% component of the overall evaluation.
- The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Minimum10experiments are suggested in the laboratory related to course content.

E. Course Outcomes:

After completion of the course, 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.

F. 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	2	3	3	1	-	-	-	-	-	-	-	2	-
CO5	1	-	1	-	-	-	-	-	-	-	-	2	1	1
CO6	3	1	-	-	-	-	-	-	-	-	-	-	1	-

G. 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://www.stanford.edu/class/cs161/
  2. http://www.itl.nist.gov/div897/sqg/dads/
  3. http://highered.mcgraw-hill.com/sites/0073523402/