CSE11110	Design and Analysis of Algorithms	L	T	P	C
Version 1.0	Contact Hours 45	3	0	0	3
Pre-requisite/Exposure	Discrete Mathematics				
Co-requisite	Concepts on Programming, Logical Ability, Problem Solving				

Course Objectives:

1. To introduce problem solving approach through design.

2. To develop students to analyse the existing algorithms and approach for improvement.

3. To introduce the students a perspective to different design and analysis approach for algorithm(s) tosolve a problem.

4. To develop students to select optimal solution to a problem by choosing the most appropriate algorithmic method.

Course Outcomes:

On the completion of this course the student will be able to

CO1: **Understand** the basics about algorithms and learn how to analyse and design algorithms

CO2: **Choose** brute force, divide and conquer, dynamic programming and greedy techniques methods tosolve computing problems

CO3: **Understand** the approach for solving problems using iterative method.

CO4: **Describe** the solution of complex problems using backtracking, branch and bound techniques.

CO5: **Classify** the different Computability classes of P, NP, NP-complete and NP-hard.

Course Description:

Algorithmic study is a core part of Computer Science. This study caters to all possible applicable areas of Computer Science. This study includes observation, design, analysis and conclusion. Various types of algorithms havedifferent notion of implementation according to their cost (in terms their time and space complexity). This study also includes refinement of one algorithm as per the applicability to real problems. Categorization of algorithms according to different method of design also includes in this course. It also compares the same algorithm using different algorithmdesign methods. For example, Knapsack problem can be solved in Greedy approach and Dynamic approach, both are optimization method. This course enables the students to think analytically while applying, designing an algorithm to solve a specific problem.

Course Content:

Unit-I	**09 Lecture Hours**
Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behaviour; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem Algorithm Design Paradigms.	
Unit-II	**09 Lecture Hours**
Sorting Algorithms & Data Structures: Selection sort, bubble sort, insertion sort, Sorting in linear time, count sort, Linear search, **Divide & Conquer:** Quick sort, worst and average case complexity, Merge sort, Matrix multiplication Binary search, Binary search tree, Strassen's algorithm for matrix multiplication, The substitution method for solving recurrences, The recursion-tree method for solving recurrences, The master method for solving recurrences.	
Unit-III	**09 Lecture Hours**
Greedy algorithms: General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm- Knapsack Problem **Dynamic programming:** Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming- Making Change Problem, Assembly Line Scheduling, Knapsack problem, Matrix chain multiplication, Longest Common Subsequence Dynamic Programming using Memoization.	
Unit-IV	**09 Lecture Hours**
Graph Algorithms : Representations of graphs, Breadth-first search, Depth first-search, Topological sort, Strongly connected components, Minimum Spanning Trees, Growing a minimum-spanning tree, The algorithms of Kruskal and Prim, Single-Source Shortest Paths, Bellman-Ford algorithm, Single- source shortest paths in directed acyclic graphs, Dijkstra's algorithm, Difference constraints and shortest paths, Proofs of shortest-paths properties, All-Pairs Shortest Paths, Shortest paths and matrix multiplication, The Floyd-Warshall algorithm, Johnson's algorithm for sparse graphs, Maximum Flow, Flow-networks, The Ford-Fulkerson method, **Branch & Bound & Backtracking**	
Unit-V	**09 Lecture Hours**
String Matching The naive string-matching algorithm, The Rabin-Karp algorithm, String matching with finite automata, The Knuth-Morris-Pratt algorithm **Approximation Algorithms:** The vertex-cover problem, The traveling-salesman problem, The set-covering problem, Randomization and linear programming **NP-Completeness:** Polynomial time, Polynomial-time verification, NP-completeness and reducibility, NP-completeness proofs , NP-complete problems.	
Text Books: 1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest And Clifford Stein, MIT Press/ Mcgraw-Hill. 2. Fundamentals of Algorithms– E. Horowitz Et Al. 3. Algorithm Design, 1ST Edition, Jon Kleinberg and Évatardos, Pearson. 4. Book 3– Author– Publisher **Reference Books:** 1. Algorithm Design: Foundations, Analysis, And Internet Examples, Second Edition, Michael T Goodrich And Roberto Tamassia, Wiley. 2. Algorithms -- A Creative Approach, 3RD Edition, Udimanber, Addison-Wesley, Reading, MA.	

Modes of Evaluation: Quiz/Assignment/Presentation/Extempore/ Written Examination

Examination Scheme:

Components	Mid Term	Class Assessment	End Term
Weightage (%)	**20**	**30**	**50**

Relationship between the Course Outcomes (COs) and Program Outcomes (POs)

Mapping between COs and POs		
Course Outcomes (COs)		**Mapped Program Outcomes**
CO1	**Understand** the basics about algorithms and learn how to analyse and design algorithms	PO1, PO, PO3, PO4, PSO1, PSO2
CO2	Choose brute force, divide and conquer, dynamic programming and greedy techniques methods to solve computing problems	PO1, PO, PO3, PO4, PSO1, PSO2
CO3	Understand the approach for solving problems using iterative method.	PO1, PO, PO3, PO4, PSO1, PSO2
CO4	Describe the solution of complex problems using backtracking, branch and bound techniques.	PO1, PO, PO3, PO4, PSO1, PSO2
CO5	Classify the different Computability classes of P, NP, NP-complete and NP-hard.	PO1, PO, PO3, PO4, PSO1, PSO2

		Engineering knowledge	Problem analysis	Design/development of solutions	Conduct investigations of complex problems	Modern tool usage	The engineer and society	Environment and sustainability	Ethics	Individual and team work	Communication	Project management and finance	Life-long learning	Adequate strong skills in learning new programming environments, analyse and design algorithms for	The ability to understand the evolutionary changes in computing, apply standard practices and strategies in	Ability to analyse the impact of Computer Science and Engineering solutions in the societal and human
Course Code	**Course Title**	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CSE 11110	Design and Analysis of Algorithms	3	3	3	3	-	-	-	-	-	-	-	-	3	3	-