Semester: 5th			
Paper code: AIDS303/AIML303	**L**	**T/P**	**Credits**
Subject: Design and Analysis of Algorithms	**4**	**0**	**4**

Marking Scheme:
1. Teachers Continuous Evaluation: As per university examination norms from time to time
2. End Term Theory Examination: As per university examination norms from time to time

INSTRUCTIONS TO PAPER SETTERS: Maximum Marks: As per university norms

1. There should be 9 questions in the end term examination question paper.
2. Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions.
3. Apart from Question No. 1, the rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, students may be asked to attempt only 1 question from each unit.
4. The questions are to be framed keeping in view the learning outcomes of course/paper. The standard/ level of the questions to be asked should be at the level of the prescribed textbooks.
5. The requirement of (scientific) calculators/ log-tables/ data-tables may be specified if required.

Course Objectives:

1.	To understand and apply the algorithm analysis techniques to generate solution space.
2.	To critically analyze the efficiency of alternative algorithmic solutions for the same problem.
3.	To analyze different algorithm design techniques.
4.	To classify a problem as computationally tractable or intractable, and discuss strategies to address intractability

Course Outcomes:

CO1	Understand the asymptotic performance of algorithms to analyze formal correctness proof for algorithms
CO2	Apply major algorithms' knowledge and data-structures corresponding to each algorithm design paradigm
CO3	Design efficient algorithms for common computer engineering design problems
CO4	Classify a problem as computationally tractable or intractable, and discuss strategies to address intractability

Course Outcomes (CO) to Programme Outcomes (PO) Mapping

(Scale 1: Low, 2: Medium, 3: High)

CO/ PO	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12
CO1	2	1	1	1	1	-	-	1	1	1	1	2
CO2	2	2	1	1	1	-	-	1	1	1	1	2
CO3	2	2	2	1	1	-	-	-	-	-	1	3
CO4	2	2	2	2	1	1	-	-	-	-	1	2

Course Overview:
This course is designed to enable the student to design and analyze algorithms for the problems. This course covers basic strategies of algorithm design: top-down design, divide and conquer, asymptotic costs, applications to sorting and searching, matrix algorithms, shortest-path and spanning tree problems, dynamic programming, greedy algorithms and graph algorithms.

UNIT I **[10]**
Introduction to Algorithms: Time Complexity and Space Complexity, Asymptotic analysis, Growth rates, some common bounds (constant, logarithmic, linear, polynomial, exponential), Complexity Analysis techniques: Master theorem, Substitution Method, Iteration Method, Time complexity of Recursive algorithms. art of problem-solving and decision making, role of data structure in algorithm design, Basic algorithmic structures of problem-solving and optimization algorithms, constraints, solution space, and feasible reasons, and representation of solution space. Sorting and searching algorithms: Selection sort, bubble sort, insertion sort, Sorting in linear time, count sort, Linear search.

UNIT II **[10]**
Divide and Conquer Algorithms: Overview of Divide and Conquer algorithms, Quick sort, Merge sort, Heap sort, Binary search, Matrix Multiplication, Convex hull and Searching, Closest Pair of Points.
Greedy Algorithms: Greedy methods with examples, Huffman Coding, Knapsack, Minimum cost Spanning trees – Prim's and Kruskal's algorithms, Single source shortest paths – Dijkstra's and Bellman Ford algorithms.

UNIT III **[10]**
Dynamic programming: Dynamic programming with examples such as Knapsack, shortest path in graph All pair shortest paths –Warshal's and Floyd's algorithms, Resource allocation problem. Backtracking, Branch and Bound with examples such as Traveling Salesman Problem, longest common sequence, n-Queen Problem.

UNIT IV: **[10]**
Graph Algorithms: Graphs and their Representations, Graph Traversal Techniques: Breadth First Search (BFS) and Depth First Search (DFS), Applications of BFS and DFS, Bipartite graphs. Graph Coloring, Hamiltonian Cycles and Sum of subsets.
Computational complexity: Problem classes: P, NP, NP-complete, NP-hard. Reduction. The satisfiability problem, vertex cover, independent set and clique problems Cook's theorem. Examples of NP-complete problems.

Textbooks:
1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", PHI ,4th Edition
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Third Edition, Pearson Education, 2006

Reference Books:

1. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Second Edition, Universities Press, 2011.
2. Anany Levitin. “Introduction to the Design and Analysis of Algorithms”, Pearson.