INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

ADVANCED ALGORITHMS								
VI Semester: CSE								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
ACSD51	**Elective**	3	0	0	3	40	60	100
Contact Classes: 48	**Tutorial Classes: Nil**	**Practical Classes: Nil**				**Total Classes: 48**		
Prerequisites: Machine Learning								

I. COURSE OVERVIEW:
This course delves into advanced concepts and techniques in algorithm design and analysis. It covers a wide range of topics that are essential for solving complex computational problems efficiently. Emphasis is placed on algorithmic paradigms, data structures, and optimization techniques that are used to address real-world challenges in computer science, engineering, and related fields.

II. COURSES OBJECTIVES:
The students will try to learn:

I. The advanced techniques for analyzing the efficiency and complexity of algorithms.
II. The graph theory concepts and algorithms for computing shortest paths.
III. The matrix-based computational methods and modular arithmetic techniques.
IV. The fundamental ideas in randomized algorithms, approximation strategies, and computational complexity theory.

III. COURSE OUTCOMES:
At the end of the course, students should be able to:

CO1 Understand algorithmic classifications, recurrence relations, and apply methods like Master's Theorem.
CO2 Analyze graph traversal techniques and shortest path algorithms, including Dijkstra's and Floyd-Warshall.
CO3 Apply greedy and dynamic programming paradigms to solve optimal graph problems and matching algorithms.
CO4 Evaluate network flow algorithms such as Ford-Fulkerson and Edmond-Karp for solving max-flow problems.
CO5 Perform matrix computations using advanced techniques like Strassen's, Gaussian Elimination, and LUP.
CO6 Analyze integer and polynomial algorithms under modulo arithmetic and assess NP-completeness in problems

IV. COURSE CONTENT:

MODULE –I: INTRODUCTION (9)

Introduction to Algorithms, Classification of Algorithms, Asymptotic Analysis, Introduction to Recurrence equations - Linear recurrences, Non-linear recurrences, Formulation of recurrence equations, techniques for solving recurrence equations, solving recurrence equations using polynomial reduction, Master's theorem

Graph: Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkstra's), depth-first search and computation of strongly connected components, Multistage Graph, topological sorting

MODULE –II: GRAPH MATCHING (10)

Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path, Bipartite matching problem Matroid**s:** Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set, Optimal tree problems- optimal merge, huffman coding, tree vertex splitting problem. Shortest Path in Graphs**:** Floyd-Warshall algorithm, Travelling Sales Person Problem and introduction to dynamic programming paradigm. Optimal Graph Problems - Minimum Spanning Tree, Single source shortest path.

MODULE-III: FLOW-NETWORKS (10)

Maxflow - minute theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm.

Matrix Computations**:** Strassen's algorithm and introduction to divide and conquer paradigm, Chain Matrix Multiplication, Matrix operations – Gaussian Elimination method, LUP-decomposition, Crout's method of decomposition, inverse of a triangular matrix,

MODULE-IV: MODULO REPRESENTATION OF INTEGERS/POLYNOMIALS (10)

Chinese Remainder Theorem, Conversion between base-representation and modulo-representation, interpolation problem. Multiplication of long integers by using Divide and Conquer paradigm, Schonhage-Strassen's Integer Multiplication algorithm.

MODULE-V: LINEAR PROGRAMMING AND NP-COMPLETENESS (9)

Linear Programming: Geometry of the feasibility region and Simplex algorithm NP-completeness: Examples, proof of NP-hardness and NP-completeness, Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures.

V. TEXTBOOKS:

1. Thomas H. Cormen, Charles E. Leiserson, Rona ld L. Rivest & Clifford Stein**, "**Introduction to Algorithms*"*, 3rd Edition, MIT Press, 2009.
2. Jon Kleinberg & Éva Tardos**,** Algorithm Design, 1st Edition, Pearson, 2005.
   Steven S. Skiena*,* The Algorithm Design Manual, 2nd Edition, Springer, 2008.

VI. REFERENCE BOOKS:

1. Alfred V. Aho, John E. Hopcroft & Jeffrey D. Ullman, "The Design and Analysis of Computer Algorithms", Pearson Education, 1974.
2. Marcello La Rocca, "Advanced Algorithms and Data Structures", Packt Publishing, 2021.

VII. ELECTRONICS RESOURCES:

1. https://www.coursera.org/learn/advanced-algorithms-and-complexity
2. https://study.iitm.ac.in/ds/course_pages/BSCS4021.html

VIII. MATERIALS ONLINE:

1. Course outline description
2. Tutorial question bank
3. Tech talk topics

4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper – II
9. Lecture notes
10. PowerPoint presentation
11. Learning Readiness Videos (ELRV)