Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER1C1	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
Advanced Algorithms	**3**	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Course Objectives: To introduce students a variety of advanced techniques, methods and results from the rapidly-developing field of algorithms to solve problems. To familiarise the state of the art in some areas of algorithmic research, including open problems.

Prerequisites: Data Structures and Algorithms.

COURSE CONTENTS

UNIT - I

Review of basic concepts; Worst case and average case analysis, Asymptotic notation, Solving recurrence equations, Medians and order statistics, Advanced data structures: Binomial Heaps, Fibonacci Heaps, Data Structures for Disjoint Sets – Disjoint-set operations, Linked-list representation of disjoint sets, Disjoint-set forests, analysis of union by rank with path compression.

UNIT - II

Advanced Design and Analysis techniques: Greedy and Dynamic Programming strategies, Backtracking, Branch and Bound. Algorithms for Knapsack problems, Matrix-Chain Multiplication problem, Traveling Salesperson Problem (TSP), etc.
Amortized analysis: the aggregate method, the accounting method, the potential method, Dynamic tables.

UNIT - III

Graph algorithms: Breadth-first search, Depth-first search, Topological sorting, Minimum Spanning Trees, Single-Source Shortest Paths, All-Pairs Shortest Paths, Maximum Flows:Augmenting Paths and Push-Relabel Methods, Minimum Cost Flows, Bipartite Matching.

UNIT - IV

Introduction to string matching problem, String matching algorithms: Naive algorithm, Rabin-Karp, Knuth-Morris-Pratt, Boyer-Moore, etc. Applications in Bioinformatics.
Computational Geometry:Convex Hull. Line-segment Intersection. Sweep Lines. Voronoi Diagrams. Range Trees. Seidel's Low-dimensional LP Algorithm.

UNIT - V

Theory of NP-Hard and NP-Complete Problems: P, NP and NP-Complete complexity classes; A few NP-Completeness proofs; other complexity classes.

Dealing with intractability: Introduction, Combinatorial Optimization, approximation factor, PTAS, FPTAS, Approximation algorithms for vertex cover, set cover, TSP, knapsack, bin packing, subset-sum problem etc. Analysis of the expected time complexity of the algorithms.

Learning Outcomes:

Upon Completing the Course, Student will have:

1. Skills to analyze algorithms
2. Comparative judgments of different design techniques
3. Ability to solve real world problems
4. Idea about the hardness of some well-known problems including TSP, vertex cover, network flow and combinatorial optimization problems.
5. Familiarity with active research areas in connection with the study of algorithms.

BOOKS RECOMMENDED:

1. T. Cormen, C. Leiserson, R. Rivest, and C. Stein. **Introduction to Algorithms**. (3rd Ed). MIT Press, McGraw-Hill, 2010.
2. M.T. Goodrich, R. Tamassia, “Algorithm design – Foundations, Analysis, and Internet Examples”, John Wiley, Second Edition.
3. V. V. Vazirani, **Approximation Algorithms**, Springer. 2001.
4. Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, **Network Flows: Theory, Algorithms, and Applications**,
5. E Horrowitz, S salmi, S Rajasekaran, “Fundamentals of Computer Algorithms”, Second Edition, University Press, 2007.
6. Aho, A V Hopcraft Ullman JD, “The Design and analysis of computer Algorithms”, Pearson Education, 2007.

LIST OF PRACTICAL ASSIGNMENTS:

Practical assignments will be based on:

1. Performance analysis
2. Solving problems using design techniques discussed
3. Solution of network flow problems
4. Approximation algorithms
5. String matching algorithms
6. Combinatorial optimization problems
7. TSP problem
8. solving some real world problems using the skill gained in the course
9. study of NP-Complete, NP-Hard problems
10. Any other emerging/ active research problems in the area in consultation with the instructor.