Devi Ahilya University, Indore, India Institute of Engineering & Technology				**III Year B.E. (Information Technology (Full Time)**			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
6ITRC2	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
Design and Analysis of Algorithms	**3**	**1**	**1**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Learning Objectives:

- Reinforce basic design concepts (e.g., pseudocode, specifications, top-down design)
- Knowledge of algorithm design strategies
- Familiarity with an assortment of important algorithms
- Ability to analyze time and space complexity

Prerequisite:

The students are required to have familiarity with the following data structures: Arrays and linked lists, Stacks and queues, Graphs and trees, binary search trees, height balancing, Heaps and priority queues

Course Objective:

The course is designed

1. To reinforce basic design concepts (e.g., pseudocode, specifications, top-down design)
2. To analyse the asymptotic performance of algorithms.
3. To demonstrate a familiarity with major algorithms and data structures.
4. To apply important algorithmic design paradigms and methods of analysis.
5. To learn and examine various Graph algorithms and their complexities.
6. To Synthesize efficient algorithms in common engineering design situations.
7. To Differentiate polynomial and nonpolynomial problems.

Course Outcome:

Students earned credits will develop ability to

CO.No.	CO	PO
CO1	Ability to write a problem's solution in pseudocode specification.	PO-1, PO2, PO3
CO2	Analyse time and space complexity and asymptotic performance of algorithms	PO-1, PO-2
CO3	Familiarity with major algorithms and data structures	PO-1, PO2, PO3
CO4	Apply important algorithmic design paradigms and methods of analysis	PO2, PO-3, PO4
CO5	Apply graph algorithms and their complexities.	PO2, PO3, PO4
CO6	Apply knowledge to synthesize efficient algorithms in common engineering design situations	PO6, PO9
CO7	Identify and differentiate polynomial and nonpolynomial problems	PO4

CO-PO Relationship

CO	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12
CO1	2	2	2									
CO2	2	2										
CO3	2	3	3									

CO4		3	3	3								
CO5		2	2	3								
CO6						3			3			
CO7				3								

* CO (rows) mention nil/very small/insignificant contribution to the PO(column)
1→ relevant and small significance 2 → medium or moderate and 3 →strong

*Note : Font type : Times New Roman
Font Size: Titles (12); Text (11)
Mail subject line : Course Obj_outcome < sub_code>_<sub_name>

COURSE CONTENTS

UNIT-I

Introduction to Algorithms: Notion of algorithms, properties, important areas of research in connection with the study of algorithms, Types of algorithms; Analysis- best case, worst case, and average case. Performance issues - Time and space complexity; Asymptotic analysis. Mathematical preliminaries; functions & their growth rates; Recurrence relations, Methods for solving recurrences.

UNIT-II

Selected Algorithms for Sorting, Searching and matrix multiplication: Elementary sorting techniques: Selection, Bubble, and Insertion sorts; Advanced sorting techniques: Heap, Merge and Quick sorts; Radix & Bucket sorts. Searching techniques: Linear and binary search; Searching minimum and maximum elements. Divide–and-Conquer strategy, Strassen's matrix multiplication

UNIT-III

Greedy Method and Dynamic Programming: Algorithms design techniques based on Greedy Method and Dynamic programming. Illustration of these strategies using appropriate examples including Knapsack problem, optimal storage on tapes, finding shortest path, all pairs shortest path, finding minimum cost spanning trees, and Matrix chain multiplication problem.

UNIT-IV

Backtracking, Branch-and- Bound, and String Matching: Backtracking and Branch-and-Bound algorithm design techniques, Illustration of these techniques using appropriate examples like Queens Problem, subset sum problem, traveling salesperson problem, etc.Introduction to string matching problem, Applications, String matching algorithms: Naive algorithm, Rabin-Karp, Knuth-Morris-Pratt, Boyer-Moore, etc.

UNIT-V

The Theory of NP-Completeness:Non-deterministic Algorithms: Introduction. Nondeterministic Complexity, Decision and optimization problems, Tractable and Intractable Problems, Computational Classes: – P, NP, NP-Complete, and NP-Hard; reducibility, Selected NP-Complete and NP-Hard problems: Hamiltonian cycle, Traveling Salesperson (TSP). Satisfiability, Clique problems, etc.

Learning Outcomes:

Students who have completed this course should be able to:

1) Have the mathematical foundation in analysis of algorithms
2) Understand different algorithmic design strategies
3) Apply design principles and concepts to algorithm design

4) Analyze the efficiency of algorithms using time and space complexity theory

Books Recommended:

[1]. Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, Introduction to Algorithms, Second Edition, MIT Press/McGraw-Hill, 2001.
[2]. Jon Kleinberg and Éva Tardos, Algorithm Design, Pearson, 2005.
[3]. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Wiley, 2006.
[4]. Knuth, D, The art of computer programming, Vols. 1-2-3, Addison Wesley 1968-73.
[5]. A V Aho, J E Hopcroft & J D Ullman, The Design and Analysis of Computer Algorithms, Addison Wesley, 1974.
[6]. E. Horowitz, S. Sahni, S Rajasekaran, Computer Algorithms, Galgotia Publications.
[7]. Saara Base, Computer Algorithms: Introduction to Design and Analysis, Addison Wesley, 2/e, 1988.
[8]. Vijay V Vazirani, Approximation Algorithms, Springer-Verlag, 2001.

.