Semester: V	
Programme : B.Sc. Computer Science (Hons)	
Course : DESIGN AND ANALYSIS OF ALGORITHMS	
Paper code: C3CS230512T / C3CS230512P	**Credits: 4**
Hours/week : Theory: 3 / Practical 2	
Category: Core/MDC/SEC/VAC : Core	
Theory / Practical / Composite : Composite	
No of Modules : 1	

Course Overview: This course introduces algorithm design and analysis, covering complexity analysis, core design techniques, sorting, tree-based data structures, and string processing. It also provides an introduction to computational complexity and NP-completeness, helping students understand efficient problem-solving and algorithmic limitations.

Course Outcome:

1. **Remember** fundamental concepts of algorithms, data structures, and asymptotic notations used in complexity analysis.
2. **Understand** and explain algorithm design paradigms such as divide and conquer, greedy methods, dynamic programming, and backtracking.
3. **Apply** appropriate algorithms and data structures to solve standard computational problems efficiently.
4. **Analyze** the time and space complexity of algorithms using summations and recurrence relations.
5. **Evaluate** different algorithmic solutions based on performance, correctness, and scalability.
6. **Create** efficient algorithmic solutions for moderate problem instances by selecting suitable design techniques and data structures.

Prerequisites: Discrete Structures

SYLLABUS

UNIT/Module	CONTENT	HOURS or NUMBER OF CLASSES	CO Mapping	COGNITIVE LEVEL
I.	Introduction to algorithms, design and analysis techniques, time and space complexity, asymptotic notations, summations, recurrences	**5**	CO1, CO4	K1, K2, K4
II.	Algorithm design techniques: Divide and conquer (Strassen's method), Greedy (Make Change), Dynamic programming (Bellman-Ford), Backtracking (8 Queens)	**8**	CO2, CO3	K2, K3, K4
III.	Sorting algorithms: Merge sort, Quick sort, average and worst-case analysis, selection problem, median and order statistics	**6**	CO3, CO4, CO5	K3, K4, K5
IV.	Generalized tree algorithms: Binary tree, Threaded binary tree, BST, AVL tree, B and B+ trees, 2–3 tree, Heap, Binomial heap	**8**	CO1, CO3, CO5	K1, K3, K5

V.	String processing: String matching, Brute force technique, KMP algorithm	**7**	CO2, CO3	K2, K3
VI.	Introduction to NP-completeness: P class, NP-hard, NP-complete, Circuit satisfiability problem	**5**	CO1, CO6	K1, K2, K6

Text Books

1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, PHI.
2. E. Horowitz , S. Sahani, R Sanguthevar, Fundamentals of Computer Algorithms, Galgotia.

Suggested readings

1. Sarabasse & A.V. Gelder Computer Algorithm – Introduction to Design and Analysis, Pearson

Web Resources

1. Introduction to Algorithms and Analysis. Sourav Mukhopadhyay. IIT Kharagpur https://nptel.ac.in/courses/106105164

Evaluation	Theory CIA: 12 Attendance: 3 Semester Exam: 45	Practical CA: 38 Attendance: 2

Paper Structure for Theory Semester Exam Module : Answer 3 out of 5 of 15 marks each

Course outcomes (COs) and Cognitive Level Mapping

COs	CO Description	Cognitive levels
CO1	Remember fundamental concepts of algorithms, data structures, and asymptotic notations used in complexity analysis.	K1
CO2	Understand and explain algorithm design paradigms such as divide and conquer, greedy methods, dynamic programming, and backtracking.	K2
CO3	Apply appropriate algorithms and data structures to solve standard computational problems efficiently.	K3
CO4	Analyze the time and space complexity of algorithms using summations and recurrence relations.	K4
CO5	Evaluate different algorithmic solutions based on performance, correctness, and scalability.	K5
CO6	Create efficient algorithmic solutions for moderate problem instances by selecting suitable design techniques and data structures.	K6