2302AS301	DESIGN AND ANALYSIS OF ALGORITHMS	L	T	P	C
		3	1	0	4

PREREQUISITE: 2301GE202 -DATA STRUCTURE CONCEPTS

COURSE OBJECTIVES:

1.	To analyse various algorithms mainly for time and space complexity.
2.	To develop algorithm for solving various computational problems by applying various algorithm design strategies.
3.	To understand the effect of choice of data structures on the complexity of algorithm.

COURSE OUTCOMES:

Upon successful completion of the course, students will be able to

CO1	Explain the concept of algorithmic efficiency, describe various asymptotic notations, and interpret recurrence relations to understand the time complexity of non-recursive and recursive algorithms.
CO2	Implement fundamental sorting and searching algorithms and apply the Divide and Conquer strategy to solve problems like Strassen's matrix multiplication.
CO3	Apply Greedy and Dynamic Programming paradigms to solve optimization problems such as the Fractional Knapsack, Minimum Spanning Tree, and All-Pair Shortest Path problems.
CO4	Design solutions for complex combinatorial problems like the N-Queen problem and the Traveling Salesperson Problem using Backtracking and Branch and Bound techniques.
CO5	Classify computational problems into complexity classes (P, NP, NP-Complete) and evaluate the theoretical lower bounds for various algorithmic problems.

COURSE CONTENTS:

Module-I	**Basic Concepts of Algorithms**	**9+3 Hours**

Notion of Algorithm, Fundamentals of Algorithmic Solving, Important problem types, Fundamentals o Asymptotic Notations and Basic Efficiency Classes, Mathematical analysis of non-recursive algorithms. Mathematical analysis of recursive algorithm: recurrence relations, solution of recurrence relations using substitution method, hashing technique.

Module-II	**Brute Force, Divide and Conquer Strategy**	**9+3 Hours**

Selection sort, Bubble sort, Sequential searching (Linear Search), Brute force string matching, General method, Merge sort, Quick Sort, Binary Search, Strassen’s matrix multiplication.

Module-III	**Greedy Approach and Dynamic Programming**	**9+3 Hours**

Fractional Knapsack problem, Minimum cost spanning tree: Prim’s and Kruskal’s algorithm, Single source shortest path problem, Principle of optimality, Multi-stage graph problem, all pair shortest path problem, 0/1 Knapsack problem, Traveling salesperson problem.

Module-IV	**Backtracking and Branch and Bound**	**9+3 Hours**

General method backtracking, N-Queen problem, Knight’s Tour Problem, General method of branch & bound, Fractional vs 0/1 knapsack problem, Traveling sales person problem using branch & bound.

Module-V	**Lower Bound Theory and Complexity Classes**	**9+3 Hours**

Lower bounds, Decision trees, P, NP and NP Complete problems, Polynomials and FFT, Approximation Algorithm, Amortized analysis, Heaps.

TOTAL: 45+15 = 60 HOURS

Mode of Assessment: Activity/CAT/ESE

COs Vs POs & PSOs MAPPING:

CO#	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2	2	1	-	-	-	-	-	-	1	1	-	-
CO2	2	3	3	3	2	-	-	-	-	-	1	2	-	-
CO3	2	3	3	3	2	-	-	2	2	-	1	2	-	-
CO4	2	3	2	3	2	-	-	2	2	1	2	3	-	-
CO5	2	**3**	3	2	1	-	-	-	-	-	**2**	**2**	-	-
*	**2**	**3**	**3**	**3**	2	**-**	**-**	**2**	**2**	**1**	**2**	**2**	**-**	-

TEXT BOOKS:

1. Algorithm Design, Jon Kelinberg and Eva Tardos, 1st Edition, Pearson Education 2014

2. Design & Analysis of Algorithms, Gajendra Sharma, Khanna Book Publishing 2018.

3. Fundamentals of algorithms, Horowitz E, Sahini S, Rajasekaran S., University Press 2008

REFERENCES:

1. Introduction to algorithms, Cormen, Leiserson, Rivest, Stein, 3rd Edition, PHI. 2012

2. An introduction to analysis of algorithms, R. Sedgewick, 1st edition, Pearson Education 1996

3. Data Structures and Program Design in C, Robert L. Kruse, C.L. Tondo, Bruce Leung, Pearson Education. 2007

4. https://archive.nptel.ac.in/courses/106/106/106106131/

5. https://archive.nptel.ac.in/courses/106/101/106101060/

6.https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/pages/syllabus/