Algorithm Design and Analysis	Code & No:	CS 305
	Credits:	3 (3,0,1)
	Pre-requisite:	CS 210
	Co-requisite:	None
	Level:	7

Course Description:

The purpose of this course is to learn several fundamental principles of algorithm design and analysis techniques. You will learn the divide-and-conquer design approaches, fast sorting, searching, and multiplication. You will learn fundamental algorithms on graphs, such as how to find shortest paths, and how to explore graphs. You will also learn practical algorithms on important data structures such as: binary search trees and heaps. Finally, you will learn about NP-Complete problems, whose status is unknown, or no polynomial-time algorithm has been discovered to solve such kind of problems.

Course Aims:

1. To create analytical skills, to enable the students to design algorithms for various applications, and to analyze the algorithms.
2. To apply design and analysis techniques to numeric and nonnumeric algorithms which act on data structures.
3. To study methods those are used to predict the resources needed by an algorithm. Specific attention is paid to worst case running time. Average-case running time is also considered.
4. To gain sophistication in the use of data structures and choice of algorithms which will enable them to better implement solutions to problems in many areas of computer science.

Student Outcomes (SOs):

☒(a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

☒(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

☒(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

☐(d) An ability to function effectively on teams to accomplish a common goal

☐(e) An understanding of professional, ethical, legal, security and social issues and responsibilities

☐(f) An ability to communicate effectively with a range of audiences

☐(g) An ability to analyze the local and global impact of computing on individuals, organizations, and society

☐(h) Recognition of the need for and an ability to engage in continuing professional development

☒(i) An ability to use current techniques, skills, and tools necessary for computing practice.

☒(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. [CS]

☒(k) An ability to apply design and development principles in the construction of software systems of varying complexity. [CS]

☐(j) An ability to use and apply current technical concepts and practices in the core information technologies of human computer interaction, information management, programming, networking, and web systems and technologies. [IT]

☐(k) An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation, and administration of computer-based systems. [IT]

☐(l) An ability to effectively integrate IT-based solutions into the user environment. [IT]

☐(m) An understanding of best practices and standards and their application. [IT]

☐(n) An ability to assist in the creation of an effective project plan. [IT]

Course Learning Outcomes (CLOs):

1. Understand fundamental computer algorithms and how to analyze them.
2. Apply the basic techniques for algorithm analysis.
3. Use the mathematical techniques required to prove the time complexity of a program/algorithm.
4. Perform inductive proofs and Prove and apply the Master Theorem.
5. Use, compare and analyze the primary sorting and searching algorithms and graph processing algorithms.
6. Recognize problems where dynamic programming is an appropriate solution method and be able to apply it.
7. Describe how to prove the correctness of an algorithm
8. Apply algorithmic complexity principles in the design of programs.

SOs and CLOs Mapping:

CLO/SO	a	B	c	d	e	f	g	h	i	j	k	l	m	n
CLO1	√	√												

CLO2	√	√											
CLO3	√	√							√	√			
CLO4	√	√							√	√			
CLO5	√	√							√	√			
CLO6	√	√	√						√	√	√		
CLO7		√								√			
CLO8			√							√	√		

No.	Topics	Weeks	Teaching hours
1	Introduction to Algorithms	1	3
2	Asymptotic Analysis	2	6
3	Divide & Conquer Algorithms (Mergesort, Quicksort, Heapsort, Recurrences, Master Theorem)	3	9
4	Linear Time Algorithms	1	3
5	Data Structures (BST, Red-Black Trees)	2	6
6	Dynamic Programming & Greedy Algorithms	2	6
7	Graph Algorithms (Graph implementation, BFS, DFS, MST, Dijkstra's Algorithm, Prim's Algorithm)	2	6
8	NP-Completeness	1	3
	Total	**14**	**42**

Textbook:

- Cormen, Thomas H. Introduction to algorithms. New Delhi: PHI Learning Private Ltd, 2010.

Essential References:

- Introduction to The Design and Analysis of Algorithms. Ananylevitin, Pearson Education, 3rd Edition,2011.
- Introduction to Design & Analysis of Algorithms, Anany Levitin, Addison Wesley, 2011.
- Foundations of Algorithms (4e). Richard E. Neapolitan And Kumarssnaimipour,." Jones And Bartlett, 4th Edition, 2009.