CS315: Advanced Data Structures	L	T	P	Data Structures
	3	1	0	

Course Objective: To study concepts of some advanced data structures like advanced trees and heaps.

S. No.	Course Outcomes (CO)
CO1	Understand and perform operations on advanced tree structures such as B-trees and Red-Black trees.
CO2	Implement and analyze mergeable heaps, including binomial and Fibonacci heaps, and their operations.

CO3	Apply fundamental graph theory concepts and definitions to analyze graph properties and components.
CO4	Implement and evaluate graph algorithms for connectivity, shortest paths, and network flows.
CO5	Design and utilize advanced data structures like tries, suffix trees, and spatial trees for efficient data handling.

S. No	Contents	Contact Hours
UNIT 1	Review of Elementary data structures- Binary Trees, Binary Heap, Sorting & Searching Technique Sparse matrices- Properties of Sparse Matrices, Linked List representation of Sparse Matrices, Analysing Algorithm ,Hashing ,Universal Hashing ,Perfect Hashing	**6**
UNIT 2	Advanced Data Structures: data structures for combinatorial Optimization – Binomial Heap, Fibonacci Heaps, Red -Black Trees, Augmenting Red – Black Trees to Dynamic order Statics and Interval Applications. Operations on Disjoint Sets –find union problem , Implementing Sets, Dictionaries, Self-Adjusting Trees, Skip lists.	**8**
UNIT 3	Divide and Conquer approach- Application of Divide and Conquer- Finding Maximum and Minimum, Finding K-th Smallest element order statistics, Finding K-th Smallest element, Merge Sort, Randomized Quick Sort.	**8**
UNIT 4	Graph Algorithms: Definitions for Graph, Algorithms for connectedness, Finding All Spanning Trees in a Weighted Graph and Planarity Testing, Breadth First and Depth First Search, Articulation Point, Cut Edge, Topological Sort, Strongly Connected Components and Single Source Shortest Path and All pair shortest path algorithms, Planer graphs.	**10**
UNIT 5	Greedy Method and Dynamic Programming: General method, knapsack problem, Single Source Shortest path, Job Sequencing with deadline, Scheduling problems. Dynamic Programming: General method,0/1 knapsack problem, All Pair Shortest Path.	**8**
UNIT 6	Advanced Algorithms: NP Complete problems, Approximation algorithms for NP complete problem (vertex cover, traveling salesman), Algorithms for matching, Flow and circular problems, Bio Inspired Algorithm- Genetic Algorithm, Particle Swam, Artificial Bee Colony, Firefly Algorithm, Bat Algorithm.	**8**
	Total	**48**