23CS21P1 - ADVANCED DATA STRUCTURES & ALGORITHM ANALYSIS LAB

(Common to CSE, CSE (AI&ML), AI&DS, and IT)

Course Category:	Professional Core	**Credits:**	1.5
Course Type:	Practical	**Lecture-Tutorial-Practical:**	0-0-3
Prerequisite:	Data Structures, Algorithms, and Strong programming skills in at least one high-level language	**Sessional Evaluation:** **Univ. Exam Evaluation:** **Total Marks:**	30 70 100
Objectives:	• To acquire practical skills in constructing and managing Data structures. • To apply the popular algorithm design methods in problem-solving scenarios.		

Course Outcomes	Upon successful completion of the course, the students will be able to:	
	CO1	Design and develop programs to solve real-world problems with popular algorithm design methods. (L5)
	CO2	Demonstrate an understanding of Non-Linear data structures by developing implementing the operations on AVL Trees, B-Trees, Heaps and Graphs. (L2)
	CO3	Critically assess the design choices and implementation strategies of algorithms and data structures in complex applications. (L5)
	CO4	Utilize appropriate data structures and algorithms to optimize solutions for specific computational problems. (L3)
	CO5	Compare the performance of different of algorithm design strategies (L4)
Course Content	**<u>Sample Programs</u>** 1. Construct an AVL tree for a given set of elements which are stored in a file. And implement insert and delete operation on the constructed tree. Write contents of tree into a new file using in-order. 2. Construct B-Tree an order of 5 with a set of 100 random elements stored in array. Implement searching, insertion and deletion operations. 3. Construct Min and Max Heap using arrays, delete any element and display the content of the Heap. 4. Implement BFT and DFT for given graph, when graph is represented by a) Adjacency Matrix b) Adjacency Lists 5. Write a program for finding the bi-connected components in a given graph. 6. Implement Quick sort and Merge sort and observe the execution time for various input sizes (Average, Worst and Best cases). 7. Compare the performance of Single Source Shortest Paths using Greedy method when the graph is represented by an adjacency matrix and adjacency lists. 8. Implement Job sequencing with deadlines using Greedy strategy. 9. Write a program to solve 0/1 Knapsack problem Using Dynamic Programming. 10. Implement N-Queens Problem Using Backtracking. 11. Use Backtracking strategy to solve 0/1 Knapsack problem. 12. Implement Travelling Sales Person problem using Branch and Bound approach.	

Text Books & References Books	**REFERENCE BOOKS:** 1. Fundamentals of Data Structures in C++, Horowitz Ellis, Sahni Sartaj, Mehta, Dinesh, 2ndEdition, Universities Press 2. Fundamentals of Algorithms - Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2ndEdition, University Press 3. Data Structures and program design in C, Robert Kruse, Pearson Education Asia 4. An introduction to Data Structures with applications, Trembley& Sorenson, McGraw Hill
E-Resources	1. http://cse01-iiith.vlabs.ac.in/ 2. http://peterindia.net/Algorithms.html