LOK JAGRUTI UNIVERSITY (LJU)

L J INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Engineering

Master of Engineering (M. E)

Semester: I

Branch: Computer Engineering (Software Engineering)

Course Code:	**20-CE-PG-049010102**
Course Name:	**Advanced Data Structures**
Category of Course:	Core
Prerequisite Course:	UG level course in Data Structures

Teaching Scheme				
Lecture (L)	**Tutorial (T)**	**Practical (P)**	**Credit**	**Total Hours**
3	**0**	**2**	**4**	**40**

Course Objectives	
1	The student should be able to choose appropriate data structures.
2	Students should be able to understand the necessary mathematical abstraction to solve problems.
3	Familiarize with algorithmic techniques such as brute force, greedy, and divide and conquer.
4	Apply advanced abstract data type (ADT) and data structures in solving real world problems.
5	To get accustomed with various programming constructs such as divide-and-conquer and dynamic programming.
6	To learn new techniques for solving specific problems more efficiently and for analyzing space and time requirements.
7	Analyze and apply graph data structure to real-life problems

Syllabus

Unit No.	Topic	Prerequisite Topic	Teaching Hours
01	**Dictionaries**		**3 (7%)**
	1.1 Definition	**---**	
	1.2 Dictionary Abstract Data Type		
	1.3 Implementation of Dictionaries		
02	**Hashing**		**5 (8%)**
	2.1 Review of Hashing, Hash Function	**---**	
	2.2 Collision Resolution Techniques in Hashing		
	2.3 Separate Chaining, Open Addressing		
	2.4 Linear Probing, Quadratic Probing		
	2.5 Extendible Hashing		
03	**Skip Lists**		**3 (10%)**
	3.1 Need for Randomizing Data Structures and Algorithms	**---**	
	3.2 Search and Update Operations on Skip Lists		
	3.3 Probabilistic Analysis of Skip Lists		
	3.4 Deterministic Skips		
04	**Trees**		**6 (15%)**
	4.1 Binary Search Trees	**---**	
	4.2 AVL Trees		
	4.3 Red Black Trees, 2-3 Trees		
	4.4 B-Trees, Splay Trees		
05	**Computational Geometry**		**4 (15%)**
	5.1 Constructing a Priority Search Tree	**---**	
	5.2 Searching a Priority Search Tree		
	5.3 Quadtrees		
	5.4 k-D Trees		
06	**Heap-1**		**3 (7%)**
	6.1 Balanced Search Trees as Heaps, Array-Based Heaps	Binary Search Tree (20-CE-PG-101-Unit-3)	
	6.2 Heap-Ordered Trees and Half Ordered Trees		
	6.3 Leftist Heaps, Skew Heaps		
07	**Heap-2**		**4 (8%)**
	7.1 Binomial Heaps	**---**	
	7.2 Changing Keys in Heaps		
	7.3 Multidimensional Heaps		
	7.4 Heap-Related Structures with Constant-Time Updates		
08	**Text Processing**		**4 (12%)**
	8.1 Sting Operations, Brute-Force Pattern Matching	**---**	
	8.2 The Boyer- Moore Algorithm		
	8.3 The Knuth-Morris-Pratt Algorithm		
09	**Tries and Dynamic Programming**		**5 (13%)**
	9.1 Standard Tries	**---**	

	9.2 Compressed Tries, Suffix Tries		
	9.3 The Huffman Coding Algorithm		
	9.4 The Longest Common Subsequence Problem (LCS)		
	9.5 Applying Dynamic Programming to the LCS Problem		
10	**Recent Trends**		**3 (10%)**
	10.1 Recent Trends in Hashing	**---**	
	10.2 Trees		
	10.3 Various computational geometry methods for efficiently solving the new evolving problem		

Course Outcome	
1	Understand the implementation of symbol table using hashing techniques
2	Develop and analyze algorithms for red-black trees, B-trees and Splay trees.
3	Develop algorithms for text processing applications
4	Identify suitable data structures and develop algorithms for computational geometry problems.
Suggested Reference Books	
1	Data Structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson, 2nd Edition,2004
2	Algorithm Design, M T Goodrich, Roberto Tamassia, John Wiley, 2002
3	Advanced Data Structures, Peter Brass, Cambridge University Press, 1st Edition
4	Data Structures and Algorithms, Narasimha Karumanchi,5th Edition
5.	Algorithms and complexity, H. S. Wil, Prentice hall.
6	Introduction to Algorithms, T. H. Cormen, C. E. Leiserson, R. L. Rivest, Prentice hall

Proposed Evaluation Scheme by Academicians (Percentage of Weightage out of 100%)					
Theory Descriptive Test		**MCQ Test**		**Hands on Project**	
Formulas and Derivation Test		**Numerical Test**		**Seminar**	

Practical Project/Hands On Project

Sr. No.	List of Practical Projects	Linked with Unit
1	Explain Dictionary as an Abstract Data Type. Implement Dictionary using suitable Data Structure.	Unit 1
2	Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client‘s telephone number	Unit 2
3	Write a program which creates Skip Lists. Implement Insert, Search and Update Operations in Skip-Lists.	Unit 3
4	Write a program which creates Binary Search Tree. And also implement recursive and non recursive tree traversing methods in order, preorder and post-order for the BST.	Unit 4
5	A Dictionary stores keywords & its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/ Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Height balance tree and find the complexity for finding a keyword	Unit 4
6	Write a program which creates Priority Search Tree. Implement Insert and Search Operations in this Tree.	Unit 5
7	Given a heap, to implement Leftist Heap using java.	Unit 6
8	Implement a program for String Matching using Boyer-Moore Algorithm on a text file content	Unit 8
9	Implement a program for String Matching using Knuth-Morris-Pratt Algorithm on a text file content.	Unit 8
10	Implement Longest Common Subsequence(LCS) Problem using Dynamic Programming Method. Show the DP table and also find the particular solution of given strings.	Unit 9

List of Recommended MOOC Courses:

1) **https://www.coursera.org/learn/advanced-algorithms-and-complexity?**
2) **https://www.edx.org/course/c-advanced-data-structures**
3) **https://onlinecourses.nptel.ac.in/noc21_cs21/preview**
4) **https://www.coursera.org/learn/advanced-data-structures?**