

## **B.TECH (CSE) II YEAR I-SEM**

<b>CS2101</b>	<b>Data Structure &amp; Algorithms</b>	<b>3L: 0T: 0P</b>	<b>3 Credits</b>
---------------	----------------------------------------	-------------------	------------------

<b>Pre-requisites</b>	<b>CS1101</b>
-----------------------	---------------

### **Course Objectives:**

1. To impart the basic concepts of data structures and algorithms.
2. To understand concepts about searching and sorting techniques
3. To understand basic concepts about stacks, queues, lists, trees and graphs.
4. To enable them to write algorithms for solving problems with the help of fundamental data structures

### **Detailed Contents:**

#### **Module 1:**

**Introduction:** Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc. Analysis of an algorithm, Asymptotic Notations, Time-Space trade off.

#### **Module 2:**

**Stacks and Queues:** ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation - corresponding algorithms and complexity analysis. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each types of Queues: Algorithms and their analysis.

#### **Module 3:**

**Linked Lists:** Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: all operations their algorithms and the complexity analysis.

#### **Module 4:**

**Searching:** Linear search and binary search techniques and their complexity analysis.

**Sorting and Hashing:** Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.

#### **Module 5:**

**Trees:** Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis, Applications of Binary Trees, B Tree, B+ Tree: definitions, algorithms and analysis, Red-Black Trees and their operations.

**Graph:** Basic Terminologies and Representations, Graph search and traversal algorithms and Complexity analysis.

### **Suggested Books:**

1. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, computer science Press.

### **Suggested Reference Books:**

1. "Algorithms, Data Structures, and Problem Solving with C++", Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company.
2. "How to Solve it by Computer", 2nd Impression by R.G. Dromey, Pearson Education.

**Course Outcomes:**

1. For a given algorithm student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness.
  2. For a given Search problem (Linear Search and Binary Search) student will able to implement it.
  3. For a given problem of Stacks, Queues and linked list student will able to implement it and analyze the same to determine the time and computation complexity.
  4. Student will able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of Space and Time complexity.
  5. Student will able to implement Graph search and traversal algorithms and determine the time and computation complexity.
-

CS2701	Data Structures & Algorithms Lab	0L: 0T: 4P	2 Credits
--------	----------------------------------	------------	-----------

Pre-requisites	
----------------	--

### **COURSE OBJECTIVE:**

This lab course is intended to write and execute programs in C to solve problems using data structures such as linked lists, stacks, queues, trees, graphs, hash tables search trees, pattern matching techniques and implement various searching and sorting methods

### **SOFTWARE REQUIREMENTS:**

Turbo C / Linux

### **SYLLABUS:**

#### **LIST OF PROGRAMS**

1. Write a program that uses functions to perform the following operations on singly linked list.
  - i) Creation
  - ii) Insertion
  - iii) Deletion
  - iv) Traversal
2. Write a program that uses functions to perform the following operations on doubly linked list.
  - i) Creation
  - ii) Insertion
  - iii) Deletion
  - iv) Traversal
3. Write a program that uses functions to perform the following operations on circular linked list.
  - i) Creation
  - ii) Insertion
  - iii) Deletion
  - iv) Traversal
4. Write a program that implement stack (its operations) using
  - i) Arrays
  - ii) Pointers
5. Write a program that implement Queue (its operations) using
  - i) Arrays
  - ii) Pointers
6. Write a program that implements the following sorting methods to sort a given list of integers in ascending order
  - i) Bubble sort
  - ii) Merge sort
  - iii) Heap sort
  - iv) Quick sort
7. Write a program that use both recursive and non recursive functions to perform the following searching operations for a Key value in a given list of integers:
  - i) Linear search
  - ii) Binary search
8. Write a program to implement binary search tree
9. Write a program to implement the tree traversal methods
10. Write a program to implement AVL Tree
11. Write a program to implement the graph traversal methods
12. Write a program to implement pattern matching algorithms
13. Write a program to implement stack using queues
14. Write a program to implement queue using stacks
15. Write a program to implement tree traversal methods without using recursion

### **TEXT BOOKS:**

1. Fundamentals of data structures in C, 2 nd edition, E.Horowitz, S.Sahni and Susan Anderson Freed, Universities Press.
2. Data structures using c – A.S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/pearson education.

**REFERENCE BOOKS:**

1. Data structures: A Pseudocode Approach with C, 2<sup>nd</sup> edition, R.F.Gilberg And B.A.Forouzan, Cengage Learning.
2. Introduction to data structures in c, 1/e Ashok Kamthane.

**COURSE OUTCOMES:**

Upon successful completion of this Lab, students will be able to:

**CO 1:** Identify the appropriate data structure for given problem

**CO 2:** Analyze the time and space complexity of algorithm or program

**CO 3:** Effectively use compilers including library functions, debuggers and trouble shooting

**CO 4:** Implement the various searching and sorting techniques

**CO 5:** Compare and contrast the abstract data types and pattern matching algorithms