2/4 B.Tech. FIRST SEMESTER

CS3L2 **DATA STRUCTURES LAB** **Credits: 2**

(Common to CSE, IT, & ECM)

Required

Lecture: -- **Internal assessment: 25 marks**

Lab: 3 periods/week **Semester end examination: 50 marks**

Course context and Overview: This course allows students to understand practically the Logical andphysical representation of data, algorithms, complexity and efficiency, data Structure operations, dense lists, and matrix representations, linked lists and their different variations, string storage representation and manipulation, queues and stacks and their applications, tree structures and their different variations, graphs and Networks , sorting techniques, searching techniques

---

Prerequisites: Data Structures

Objectives:

1. To implement recursive functions.
2. To implement stack, queue, linked list, tree and graph data structures.
3. To arrange data using different sorting techniques.

Learning Outcome:

Ability to:

1. Implement the stack, queue and their applications
2. Implement the various types of linked list and their applications
3. Perform basic operations on trees and graphs and determine minimum spanning trees
4. Implement and compare time complexities of different sorting algorithms.

Exercise 1

a) Write recursive program which computes the n^{th} Fibonacci number, for appropriate values of n.

b) Write recursive program for calculation of Factorial of an integer.

Exercise 2

a. Implementation of stack operations using arrays.
b. Implementation of queue operations using arrays.

Exercise 3

a) Railroad cars numbered are as 0,1,2,---,n-1. Each car is brought into the stack and removed at any time. For instance, if n=3, we could move 0, move 1, move 2 and then take the cars out, producing 2,1,0. Implement application for the given problem.
b) Consider a payment counter at which the customer pays for the items purchased. Every time a customer finished paying for their items, he/she leaves the queue from the front. Every time another customer enters the line to wait, they join the end of the

line. Implement the application for this problem.

Exercise 4
Implementation of singly linked list

Exercise 5
Implementation of doubly linked list

Exercise 6

a) Representation of Sparse matrix.
b) Implementation of circular linked list

Exercise 7

Implement Exercise 3 (a) using linked lists.

Exercise 8
Implement Exercise 3(b) using linked lists.

Exercise 9

a) A polynomial has the main fields as coefficient, exponent in linked list it will have one more field called link to point to next term in the polynomial. If there are n terms in the polynomial then n such nodes has to be created.

Exercise 10

Implementation of binary tree: creation, insertion, deletion, traversing

Exercise 11

Implementation of Binary Search Tree operations

Exercise 12

Implementation of Graph traversals

Exercise 13

Implementation of minimum spanning tree

Exercise 14

26, 5, 77, 1 , 61, 11, 59, 15, 48, 19

1. Arrange above data set using insertion sort
2. Arrange above data set using Quick sort
3. Arrange above data set using Merge sort

Exercise 15
90, 77, 60, 99, 55, 88, 66, 32, 41, 19

a) Arrange above data set using Heap sort

Arrange above data set using Radix sort