SEMESTER-III

COURSE 6: DATA STRUCTURES

Theory Credits: 3 3 hrs/week

Course Objectives:

- The objective of the course is to make a student to implement data structures and organize and manage data, based on data structures for efficient access.

Course Outcomes:

- Identify data structures suitable to solve any specific problem.
- Identifying various data structures and their real-time applications
- Identifying the use of Time and Space Complexity.
- Implementing different sorting & searching techniques.

Syllabus

Unit-I

Introduction and Overview- Elementary Data Organization, Data Structures classification, Data Structure Operations, Algorithms: Complexity, Time-Space Tradeoff.

Preliminaries-Mathematical Notation and Functions, Algorithmic Notation, Control Structures used in algorithms, Complexity of Algorithms. Other Asymptotic Notations, Sub algorithms, Variables, Data Types.

Case Study:

1. Calculate the space complexity of a given code

```
int tot (int a, int b)
{
        int c;
        c = a + b;
        return c;
}
```

Unit-II

Arrays, Records and Pointers – Linear Arrays, Representation and Traversing Linear Arrays, Inserting and Deleting. Passing an array to function, Pointer & Arrays Multidimensional Arrays, Sparse Matrices.

Case Study:

1. Application of arrays in the real world

Unit- III

Linked Lists – Representation, Dynamic Memory Allocation, Traversing, Searching, Insertion, Deletion, Header Linked Lists, Two-Way Lists

Stacks- Stacks, Operations on stacks, Array representation of stacks, Linked List representation of stacks, Arithmetic Expressions, Polish notation, Recursion.

Case Study:

1. Linked list verses Arrays.
2. Towers of Hanoi.

Unit- IV

Queues, Linked representation of Queues, Deques, Priority Queues.

Sorting - Insertion Sort, Bubble Sort, Selection sort, Quick Sort, Merge sort, Heap Sort, Searching – Linear Search, Binary Search.

Case Study:

1. Application of Queues.
2. Comparison of sorting algorithms.

Unit- V

Trees- Binary trees, Representing and traversing binary trees, Traversal algorithms using stacks.
Binary Search Trees, Searching, Insertion and Deletion in Binary Search Trees,

Graphs- Terminology, Sequential representation of Graphs, Linked representation of Graphs, Operations on Graphs, Traversing a Graph.

Case Study:

1. Applications of Binary Tree.
2. Warshall's Algorithm.

Text books:

1. Data Structures by Seymour Lipschutz, McGraw Hill(Schaum's Outlines).
2. Data Structures using C , Second edition , Dr. Reema Thareja, Oxford University Press.

REFERENCE BOOKS:

1. Data Structures & Algorithms Using C, Khanna Publishers
2. Theory and Problems of Data Structures by Seymour Lipschutz, McGraw Hill (Schaum's Outlines)
3. Data Structures & Algorithms in C by M.A.Weiss, Addison Wisley.
4. Data Structures Using C, Reema Thareja, oxford.

SEMESTER-III

COURSE 6: DATA STRUCTURES

Practical Credits: 1 2 hrs/week

List of Lab Experiments

1. Write a C program to Implement matrix multiplication.
2. Write a C program to Implement stack using arrays.
3. Write a C program to Implement queue using arrays.
4. Write a C program to Implement circular queue using arrays.
5. Write a C program to Implement dequeue using arrays.
6. Write a C program to Implement single linked list using the methods create(), insert(), search(), delete() and display().
7. Write a C program to Implement double linked list.
8. Write a C program to Implement stack using linked list.
9. Write a C program to Implement queue using linked list.
10. Give a solution to towers of Hanoi using C program.
11. Write a C program to Implement bubble sort.
12. Write a C program to Implement selection sort.
13. Write a C program to Implement insertion sort.
14. Write a C program to Implement merge sort.
15. Write a C program to Implement quick sort.