DATA STRUCTURES AND ALGORITHMS

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

Course Code: 22CT1103

L	T	P	C
3	0	0	3

COURSE OUTCOMES:

At the end of the Course the student shall be able to

CO1: apply logarithmic analysis and describe various searching and sorting techniques. (L3)

CO2: describe the concepts of stacks and queues. (L2)

CO3: apply the concepts of linked lists. (L3)

CO4: demonstrate the concepts of trees. (L3)

CO5: illustrate the concepts of graphs. (L3)

UNIT-I — 10 Lectures

Analysis of Algorithms: Efficiency of algorithms, Apriori Analysis, Asymptotic notations.

Searching: Introduction, linear search, binary search, Fibonacci search.

Sorting: Introduction, bubble sort, insertion sort, selection sort, quick sort, merge sort.

Learning Outcomes: At the end of the unit, the student will be able to

1. illustrate how linear, binary search and Fibonacci search would work with examples. (L2)
2. describe insertion, selection, and bubble sort(L2)
3. implement programs for linear, binary, and Fibonacci search using arrays. (L3)
4. implement programs for quick sort, merge sort. (L3)

UNIT-II — 10 Lectures

Abstract Data Types: Introduction, List ADT, Stack ADT, Queue ADT.

Stacks: Introduction, stack operations, applications.

Queues: Introduction, Operations on queues, circular queues, Priority queues, applications.

Learning Outcomes: At the end of the unit, the student will be able to

1. illustrate how stack and queue would work with examples. (L2)
2. describe the advantages of circular, priority queue. (L2)
3. describe the applications of stack, queue, circular queue and priority queue (L2)
4. write programs for stack and queue using arrays. (L2)

UNIT-III — 10 Lectures

Linked Lists: Introduction, singly linked lists, circular linked lists, doubly linked lists, multiple linked lists, applications.

Linked stacks and linked queues: Introduction, operations on linked stacks and linked queues,

dynamic memory management, implementation of linked representations, applications.

Learning Outcomes: At the end of the unit, the student will be able to

1. describe the advantages of linked implementation over array implementation of various data structures. (L2)
2. demonstrate how to declare structures to be used in simple linked lists, doubly linked lists and circular linked lists. (L3)
3. write the programs for inserting, deleting, and searching in a simple linked list. (L3)

UNIT-IV 10 Lectures

Trees and binary trees: Introduction, Trees: definition and basic terminologies, representation of trees. Binary trees: basic terminologies and types, representation of binary trees, binary tree traversals, applications.

Binary search trees and AVL trees: Introduction, binary search trees: definition and operations, AVL Trees: definition and operations, applications.

Heaps: Heaps, Priority Queues, Definition of a Max Heap, Insertion into a Max Heap, Deletion from a Max Heap, Applications: Heap Sort.

Learning Outcomes: At the end of the unit, the student will be able to

1. explain what is meant by a balanced binary tree and why it is important. (L2)
2. describe AVL trees, what the definition of AVL property is and demonstrate the ability to identify diagrams of trees as to whether they have the AVL property. (L2)
3. discuss, with diagrams, the algorithms for a single left rotation, single right rotation, double left rotation, and double right rotation in AVL trees. (L2)
4. demonstrate how to declare structures to be used in binary trees. (L3)
5. implement the algorithms for inserting, deleting, and searching for nodes in a binary tree. (L3)

UNIT-V 10 Lectures

GRAPHS: Introduction, definitions and basic terminologies, representations of graphs, graph traversals and applications.

Learning Outcomes: At the end of the unit, the student will be able to

1. discuss a basic search algorithm for graphs. (L2)
2. demonstrate minimal spanning tree and discuss, with diagrams, Prim's algorithm for finding the minimal spanning tree of a graph. (L3)
3. define shortest path and discuss, with diagrams, Dijkstra's algorithm for finding the shortest path from node x to node y of a graph. (L2)

TEXT BOOKS:

1. Ellis Horowitz, Sartaj Sahni, Dinesh Mehta, *Fundamentals of Data structures in C++,* 2nd Edition, University Press (India) Pvt.Ltd.,2008.

REFERENCES:

1. G.A.V. PAI, *Data Structures and Algorithms, Concepts, Techniques and Applications*, Volume1,1st Edition, Tata McGraw-Hill,2008.
2. Richard F. Gilberg & Behrouz A. Forouzan, *Data Structures, Pseudo code Approach with C*, 2nd Edition, Cengage Learning India Edition,2007.
3. Langsam, M. J. Augenstein, A. M. Tanenbaum, *Data structures using C and C++*, 2nd Edition, PHI Education,2008.
4. Sartaj Sahni, Ellis Horowitz, *Fundamentals of Data Structures in C*, 2nd Edition, Orient black swan, 2010.

WEB REFERENCES:

1. https://www.cs.usfca.edu/~galles/visualization/Algorithms.html