

Data Structure & C

Code: EC504B

Contact: 3L + 1T

Credits: 4

Pre-requisites: CS 201 (Basic Computation and Principles of C), M101 & M201 (Mathematics), basics of set theory

Module -I. Linear Data Structure [8L]

Introduction (2L):

Why we need data structure?

Concepts of data structures: a) Data and data structure b) Abstract Data Type and Data Type.

Algorithms and programs, basic idea of pseudo-code.

Algorithm efficiency and analysis, time and space analysis of algorithms – order notations.

Array (2L):

Different representations – row major, column major.

Sparse matrix - its implementation and usage. Array representation of polynomials.

Linked List (4L):

Singly linked list, circular linked list, doubly linked list, linked list representation of polynomial and applications.

Module -II: Linear Data Structure [7L]

[Stack and Queue (5L):

Stack and its implementations (using array, using linked list), applications.

Queue, circular queue, dequeue. Implementation of queue- both linear and circular (using array, using linked list), applications.

Recursion (2L):

Principles of recursion – use of stack, differences between recursion and iteration, tail recursion.

Applications - The Tower of Hanoi, Eight Queens Puzzle.

Module -III. Nonlinear Data structures [15L]

Trees (9L):

Basic terminologies, forest, tree representation (using array, using linked list).

Binary trees - binary tree traversal (pre-, in-, post- order), threaded binary tree (left, right, full) - non-recursive traversal algorithms using threaded binary tree, expression tree.

Binary search tree- operations (creation, insertion, deletion, searching).

Height balanced binary tree – AVL tree (insertion, deletion with examples only).

B- Trees – operations (insertion, deletion with examples only).

Graphs (6L):

Graph definitions and concepts (directed/undirected graph, weighted/un-weighted edges, sub-graph, degree, cutvertex/ articulation point, pendant node, clique, complete graph, connected components – strongly connected component, weakly connected component, path, shortest path, isomorphism).

Graph representations/storage implementations – adjacency matrix, adjacency list, adjacency multi-list.

Graph traversal and connectivity – Depth-first search (DFS), Breadth-first search (BFS) – concepts of edges used in DFS and BFS (tree-edge, back-edge, cross-edge, forward-edge), applications.

Minimal spanning tree – Prim's algorithm (basic idea of greedy methods).

Module - IV. Searching, Sorting [10L]

Sorting Algorithms (5L): Bubble sort and its optimizations, insertion sort, shell sort, selection sort, merge sort, quick sort, heap sort (concept of max heap, application –priority queue), radix sort.

Searching (2L): Sequential search, binary search, interpolation search.

Hashing (3L): Hashing functions, collision resolution techniques.