**UE23CS252A: Data Structures and its Applications (4-0-2-5-5)**

This course introduces abstract concepts, shows how the concepts are useful for problem solving and then shows how the abstractions can be made concrete by using a programming language. Equal emphasis is placed on both the abstract and the concrete versions of a concept so that the student learns about the concept itself, its implementation and its application.

Course Objectives:

● Basic approaches and mindsets for analyzing and designing data structures and construct essential skills of data structures to store and retrieve data quickly and usefully (efficiently).

● Usage of different data structures that support different set of operations which are suitable for different type of tasks.

● Implement how to insert, delete, search and modify data in any given data structures- Stack, Queue, List, Tree, heap, Graphs.

● Implement a given application using the available data structure.

Course Outcomes:

At the end of this course, the student will be able to,

● Choose relevant data structures for any given application

● Apply the required to implement any data structure.

● Appropriate data structure in competitive programming.

● Design and develop efficient software systems with good knowledge of data structures.

Course Content:

Unit 1: Linked List and Stacks

Review of C, Static and Dynamic Memory Allocation. Linked List: Doubly Linked List, Circular Linked List – Single and Double, Multilist: Introduction to sparse matrix (structure). Skip list Case study: Dictionary implementation using skip list. Stacks: Basic structure of a Stack, Implementation of a Stack using Arrays & Linked list. Applications of Stack: Function execution, Nested functions, Recursion: Tower of Hanoi. Conversion & Evaluation of an expression: Infix to postfix, Infix to prefix, Evaluation of an Expression, Matching of Parenthesis.

14 Hours

Unit 2: Queues and Trees

Queues & Dequeue: Basic Structure of a Simple Queue, Circular Queue, Priority Queue, Dequeue and its implementation using Arrays and Linked List. Applications of Queue: Case Study – Josephus problem, CPU scheduling- Implementation using queue (simple /circular). General: N-ary trees, Binary Trees, Binary Search Trees (BST), and Forest: definition, properties, conversion of an N-ary tree and a Forest to a binary tree. Traversal of trees: Preorder, Inorder and Postorder.

14 Hours

Unit 3: Application of Trees and Introduction to Graphs

Implementation of BST using arrays and dynamic allocation: Insertion and deletion operations, Implementation of binary expression tree, Threaded binary search tree and its implementation. Heap: Implementation using arrays. Implementation of Priority Queue using heap - min and max heap. Applications of Trees and Heaps: Implementation of a dictionary / decision tree (Words with their meanings). Balanced Trees: definition, AVL Trees, Rotation, Splay Tree, Graphs: Introduction, Properties, Representation of graphs: Adjacency matrix, Adjacency list. Implementation of graphs using adjacency matrix and lists. Graph traversal methods: Depth first search, Breadth first search techniques. Application: Graph representation: Representation of computer network topology.

14 Hours

Unit 4: Applications of Graphs , B-Trees, Suffix Tree and Hashing

Application of BFS and DFS: Connectivity of graph, finding path in a network. Suffix Trees: Definition, Introduction of Trie Trees, Suffix trees. Implementations of TRIE trees, insert, delete and search operations. Hashing: Simple mapping / Hashing: hash function, hash table, Collision Handling: Separate Chaining & Open Addressing, Double Hashing, and Rehashing. Applications: URLs decoding, Word prediction using TRIE trees / Suffix Trees.

14 Hours

Tool/ Languages: C Programming Language

Lab / Hands-on: 14 Hours

Implementation of:

1: Linked List and advanced operations.

2: Stack and applications based on it.

3: Queue and applications based on it.

4: Binary Tree, Binary Search Tree and applications based on it.

5: Graph Data structure and applications based on it.

6: Hashing Techniques.

Text Book(s):

1: "Data Structures using C / C++", Langsum Yedidyah, Moshe J Augenstein, Aaron M Tenenbaum Pearson Education Inc, 2nd edition, 2015.

Reference Book(s):

1: "Data Structures and Program Design in C", Robert Kruse, Bruce Leung, C.L Tondo, Shashi Mogalla, Pearson, 2nd Edition, 2019.