Paper – VI - GE: Data Structures and Algorithms

[4 HPW:: 4 Credits :: 100 Marks]

Objectives:

- To introduce the time and space complexities of algorithms.
- To discuss the linear and non-linear data structures and their applications.
- To introduce the creation, insertion and deletion operations on binary search trees and balanced binary searchtrees.
- To introduce various internal sorting techniques and their time complexities

Outcomes:

Students will be

- Able to analyze the time and space complexities of algorithms.
- Able to implement linear, non-linear data structures and balanced binarytrees
- Able to analyze and implement various kinds of searching and sorting techniques.
- Able to find a suitable data structure and algorithm to solve a real world problem.

UNIT–I

Performance and Complexity Analysis: Space Complexity, Time Complexity, Asymptotic Notation (Big-Oh), Complexity Analysis Examples.

Linear List-Array Representation: Vector Representation, Multiple Lists Single Array.
Linear List-Linked Representation: Singly Linked Lists, Circular Lists, Doubly Linked Lists, Applications (Polynomial Arithmetic).

Arrays and Matrices: Row and Column Major Representations, Sparse Matrices.

Stacks: Array Representation, Linked Representation, Applications (Recursive Calls, Infix to Postfix, Postfix Evaluation).

Queues: Array Representation, Linked Representation. **Skip Lists and Hashing:** Skip Lists Representation, Hash Table Representation, Application- Text Compression.

UNIT- II

Trees: Definitions and Properties, Representation of Binary Trees, Operations, Binary TreeTraversal.
Binary Search Trees: Definitions, Operations on Binary Search Trees.

Balanced Search Trees: AVL Trees, and B-Trees.

UNIT –III

Graphs: Definitions and Properties, Representation, Graph Search Methods (Depth First Search and Breadth First Search)
Application of Graphs: Shortest Path Algorithm (Dijkstra), Minimum Spanning Tree (Prim's and Kruskal's Algorithms).

UNIT –IV

Searching : Linear Search and Binary Search Techniques and their complexity analysis.
Sorting and Complexity Analysis: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, and Heap Sort.Algorithm Design Techniques: Greedy algorithm, divide-and-conquer, dynamic programming.

Suggested Reading:
1. Michael T. Goodrich, Roberto Tamassia, David M. Mount, *Data Structures and Algorithms Python* John Wiley & Sons, 2013.
2. Problem Solving with algorithams and Data Structures Using Python by Miller and David L. Ranum
3. Algorithmic Problem Solving with Python by John B. Schneider