

## **Data Structure and Algorithm**

### Course Objectives:

- 1.To provide fundamental knowledge of various data structures and their implementation
- 2.To provide the fundamental knowledge of various algorithms and their analysis

### 1. Concept of data structure (2 hours)

- a.Introduction: data types, data structures and abstract data types
- b.Introduction to algorithms

•

### 2. The Stack and Queue (6 hours)

- a.Stack operation
- b.Stack application: Evaluation of Infix, Postfix and Prefix expressions
- c.Operations in queue, Enqueue and Dequeue
- d.Linear and circular queue
- e.Priority queue

•

### 3. List (3 hours )

- a.Definition
  - i. Static and dynamic list structure
  - ii. Array implementation of lists
  - iii. Queues as list

•

### 4. Linked lists (5 hours )

- a.Dynamic implementation
- b.Operations in linked list
- c.Linked stacks and queues
- d.Doubly linked lists and its applications

•

### 5. Recursion (4 hours )

- a.Principle of recursion
- b.TOH and Fibonacci sequence
- c.Applications of recursion

•

### 6. Trees (7 hours )

- a.Concept
- b.Operation in Binary tree
- c.Tree search, insertion/deletions
- d.Tree traversals (pre-order, post-order and in-order)
- e.Height, level and depth of a tree
- f.AVL balanced trees and Balancing algorithm
- g.The Huffman algorithm
- h.B-Tree
- i.Red Black Tree

### 7. Sorting (5 hours )

- 1.Types of sorting: internal and external
- 2.Insertion and selection sort
- 3.Exchange sort
- 4.Merge and Redix sort

5. Shell sort
6. Heap sort as a priority queue
7. Big 'O' notation and Efficiency of sorting
8. Searching ( 5 hours )
  - a. Search technique
  - b. Sequential, Binary and Tree search
  - c. General search tree
  - d. Hashing
    - i. Hash function and hash tables
    - ii. Collision resolution technique
9. Growth Functions ( 2 hours )
  - a. Asymptotic notations: notations and their properties
10. Graphs ( 6 hours )
  - a. Representation and applications
  - b. Transitive closure
  - c. Warshall's algorithm
  - d. Graphs type
  - e. Graph traversal and Spanning forests
    - i. Depth First Traversal and Breadth First Traversal
    - ii. Topological sorting: Depth first, Breadth first topological sorting
    - iii. Minimum spanning trees, Prim's, Kruskal's and Round-Robin algorithms
  - f. Shortest-path algorithm
    - i. Greedy algorithm
    - ii. Dijkstra's Algorithm

#### Practical:

There shall be 10 to 12 lab exercises based on C or C++

1. Implementation of stack
2. Implementations of linear and circular queues
3. Solutions of TOH and Fibonacci sequence by Recursion
4. Implementations of linked list: singly and doubly linked list
5. Implementation of trees: AVL trees, and balancing
6. Implementation of Merge sort
7. Implementation of search: sequential, Binary and Tree search
8. Implementation of Graphs: Graph Traversals
9. Implementation of hashing
10. Implementation of Heap

#### References

1. Y. Langsam, M. J. Augenstein and A. M Tenenbaum, "Data Structures using C and C++", PHI
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, "Introduction to Algorithms", PHI
3. G.W. Rowe, "Introduction to Data Structure and Algorithms with C and C++", PHI
4. R. L. Kruse, B. P. Leung, C. L. Tondo, "Data Structure and Program design in C", PHI
5. G. Brassard and P. Bratley, "Fundamentals of Algorithms", PHI