# DATA STRUCTURE AND ALGORITHMS [CT 552] - SYLLABUS

DATA STRUCTURE AND ALGORITHMS [CT 552] - SYLLABUS

| | | | | | |
|---|---|---|---|---|---|
| Lecture | : | 3 | Year | : | II |
| Tutorial | : | 0 | Part | : | II |
| Practical | : | 3 | | | |

Course Objectives:

To provide fundamental knowledge of various data structures and their implementation
To provide the fundamental knowledge of various algorithms and their analysis

1. Concept of data structure (2 hours)

1.1 Introduction: data types, data structures and abstract data types
1.2 Introduction to algorithms

2. The Stack and Queue (6 hours)

2.1 Stack operation
2.2 Stack application: Evaluation of Infix, Postfix and Prefix expressions
2.3 Operations in queue, Enqueue and Dequeue
2.4 Linear and circular queue
2.5 Priority queue

3. List ( 3 hours )

3.1 Definition
1.1.1 Static and dynamic list structure
1.1.2 Array implementation of lists
1.1.3 Queues as list

4. Linked lists ( 5 hours )

4.1 Dynamic implementation
4.2 Operations in linked list
4.3 Linked stacks and queues
4.4 Doubly linked lists and its applications

5. Recursion ( 4 hours )

5.1 Principle of recursion
5.2 TOH and Fibonacci sequence
5.3 Applications of recursion

6. Trees ( 7 hours )

6.1 Concept
6.2 Operation in Binary tree
6.3 Tree search, insertion/deletions
6.4 Tree traversals (pre-order, post-order and in-order)
6.5 Height, level and depth of a tree
6.6 AVL balanced trees and Balancing algorithm
6.7 The Huffman algorithm
6.8 B-Tree
6.9 Red Black Tree

7. Sorting ( 5 hours )

7.1 Types of sorting: internal and external
7.2 Insertion and selection sort
7.3 Exchange sort
7.4 Merge and Redix sort
7.5 Shell sort
7.6 Heap sort as a priority queue
7.7 Big ‘O’ notation and Efficiency of sorting

8. Searching ( 5 hours )

8.1 Search technique
8.2 Sequential, Binary and Tree search
8.3 General search tree
8.4 Hashing
1.1.4 Hash function and hash tables
1.1.5 Collision resolution technique

9. Growth Functions ( 2 hours)

Asymptotic notations: , O,, o, notations and their properties

10. Graphs ( 6 hours )

10.1 Representation and applications
10.2 Transitive closure
10.3 Warshall's algorithm
10.4 Graphs type
10.5 Graph traversal and Spanning forests
1.1.6 Depth First Traversal and Breadth First Traversal
1.1.7 Topological sorting: Depth first, Breadth first topological sorting
1.1.8 Minimum spanning trees, Prim's, Kruskal's and Round-Robin algorithms
10.6 Shortest-path algorithm
1.1.9 Greedy algorithm
1.1.10 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