Course Title : Data Structure and Algorithms

Course Code: BTCS 301-18

Semester: 3rd

PTU SYLLABUS

- Module 1: Introduction Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. Searching: Linear Search and Binary Search Techniques and their complexity analysis. [6 hrs] (CO1)

- Module 2: Stacks and Queues ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation – corresponding algorithms and complexity analysis. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each types of Queues: Algorithms and their analysis. [10 hrs] (CO2, CO4, CO5)

- Module 3: Linked Lists Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: All operations their algorithms and the complexity analysis. Trees: Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. B Tree, B+ Tree: definitions, algorithms and analysis. [10 hrs] (CO2, CO4, CO5)

- Module 4: Sorting and Hashing Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing. [10 hrs] (CO3)

- Module 5: Graph Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis. [6 hrs] (CO2, CO4)

Module 1: Introduction

- Basic Terminologies:

- DATA : raw facts / set of values
- Eg : 25

- INFORMATION: useful or processed form of data.
- Eg: age=25

Basic Terminologies: contd..

- Data Item: single unit of value

Types: 1.Group Data Item

2. Elementary Data Item

- Entity
- Entity Set
- Field
- Record

Data Structure

- The logical or mathematical model of a particular organization of data is called data structure.
- It is a particular way of storing and organizing data in a computer so that it can be used efficiently.
- Types of Data Structures:
  - Linear
    - Arrays
    - Linked Lists
    - Stack
    - Queue
  - Non-Linear
    - Tree
    - Graphs

Data Structure Operations

- Traversing : Accessing each element exactly once.
- Searching : Finding the location of a particular record.
- Insertion : Adding a new record to the structure.
- Deletion : Removing a record from the structure.
- Sorting : Arranging the records in some logical order.

Data Types

- Character
- Integer
- Floating
- Logical

Algorithm

- Well defined list of steps for solving a particular problem.
- Performance Metrics:

  Time and space
- Complexity of an algorithm is the function which gives the running time and space in terms of the input size.

Characteristics Of a Data Structure

- **Time Complexity –** Running time or execution time of operation of data structure must be as small as possible.
- **Space Complexity** – Memory usage of a data structure operation should be as little as possible.

Types of complexity: Worst case

Average case

Best case

Asymptotic Notations

- O Notation : Formal way to express upper bound of an algorithm's running time.
- Ω Notation : Formal way to express lower bound of an algorithm's running time.
- ϴ Notation : Formal way to express both upper and lower bound of an algorithm's running time.

Rate of Growth

Suppose M is an algorithm, and n is the size of the input data.
Clearly the complexity f(n) of M increases as n increased.

It is usually the rate of increase of f(n) that we want to examine.

This is done by comparing f(n) with some standard functions, such as $\log_2 n$, n, $n!$, $n \log_2 n$, n^3, 2^n

2015

IMPORTANT

JANUARY

WEEK 03 | 014-349

WEDNESDAY 16

$\log_2 n$ grows most slowly
2^n grows most rapidly.

One way to compare the function f(n) with these standard functions is to use the functional O notation defined as follows:-

$8n^3 - 576n^2 + 832n - 248 = O(n^3)$

Complexities :-

1) Linear search → $O(n)$
2) Binary search → $O(\log n)$