Sunbeam Institute of Information Technology
Pune and Karad
PreCAT

Module – Data Structures

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Data Structures - Introduction

- Section B of CCAT
- Total 7 questions will be asked on data structures
- Focus will be on basic data structures algorithms
- Introduction to Advanced data structures

What is Data structure?

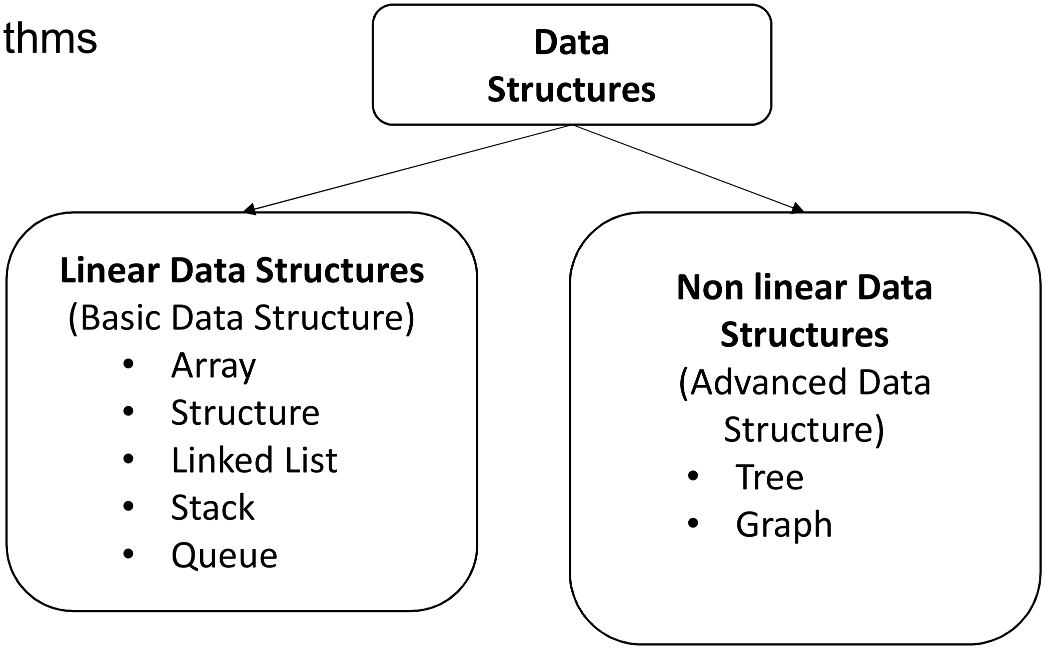
- Organising data into memory
- Processing the data efficiently

Why we need Data Structure?

To achieve

1. Efficiency
2. Reusability

There are two types of Data Structures.



Data Structures

Linear Data Structures
(Basic Data Structure)

- Array
- Structure
- Linked List
- Stack
- Queue

Non linear Data Structures
(Advanced Data Structure)

- Tree
- Graph

Programming Language

- DS and Algorithms are language independent
- We will use **C programming** to implement Data structures

Data Structures - Introduction

Linear Data Structures

- Data elements are arranged linearly (sequentially) into the memory.
- Data elements can be accessed linearly / Sequentially.

Non linear Data Structures

- Data elements are arranged in non linear manner (hierarchical) into the memory.
- Data elements can be accessed non linearly.

+ Array

- Searching Algorithms
- Sorting Algorithms

+ Linked List

- Concept and Definition
- types of Linked List
- Operations – Add, Delete, Traverse etc
- Difference between Array and Linked List

+ Stack

- Concept and Definitions
- Implementation of stack
- Stack Applications

+ Queue

- Concept and Definitions
- Types of queue
- Implementation of Linear queue and circular queue
- Queue Applications

+ Tree

- Terminologies

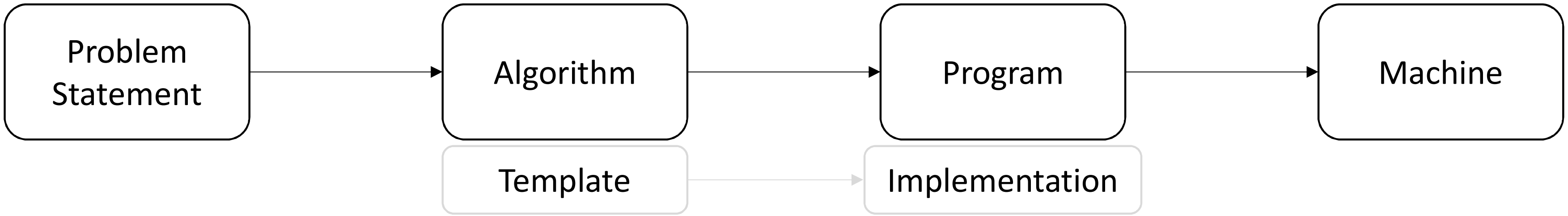
+ Graph

- Terminologies

Data Structures - Introduction

- **Algorithm**
  - is a clearly specified set of simple instructions.
  - is a solution to solve a problem.
  - is written in human understandable language.
  - Algorithm is also referred as “pseudo code”.

One problem statement has multiple solutions, out of which we need to select efficient one. Hence we need to do analysis of an algorithm.



Problem Statement → Algorithm → Program → Machine

Template → Implementation

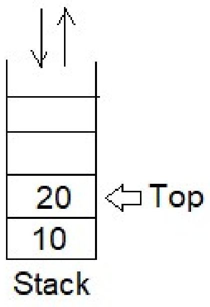
e.g. Write an algorithm to find sum of all array elements.
Algorithm:
Step 1: Initialize sum =0
Step 2: Traverse array form index 0 to N-1
Step 3: Add each element in the sum variable
Step 4: Return the final sum

```
Algorithm SumArray(array, size) {
        sum = 0;
        For(index = 0 ; index < size ; index++)
                sum += array[index];
        return sum;
}
```

Stack

Stack

- Stack is Last-In-First-Out structure.
  - Stack Operations:
    - push()
    - pop()
    - peek()
    - is_empty()
    - is_full()



Stack

- Parenthesis balancing
- Expression conversion and evaluation
- Function calls
- Used in advanced data structures for traversing

- **Expression conversion and evaluation:**
  - Infix to postfix
  - Infix to prefix
  - Postfix evaluation
  - Prefix evaluation
  - Prefix to postfix
  - Postfix to infix

Thank you!

Devendra Dhande

<devendra.dhande@sunbeaminfo.com>