Introduction

General Guideline

© (2021) ABES Engineering College.

This document contains valuable confidential and proprietary information of ABESEC. Such confidential and proprietary information includes, amongst others, proprietary intellectual property which can be legally protected and commercialized. Such information is furnished herein for training purposes only. Except with the express prior written permission of ABESEC, this document and the information contained herein may not be published, disclosed, or used for any other purpose.

Session Plan

The session will cover basic terminology and introduction to data structure:-

- Basic terminology
- Data structure
- Need for data structure
- Primitive operations
- Abstract data types
- Classification of data structure

Basic terminology

Entity and Attribute : An entity represents the class of certain objects. it contains various attributes. Each attribute represents the particular property of that entity.

Data: Data can be defined as an elementary value or the set (collection) of values, for example, student's name and its id are the data about the student.

Record: Record can be defined as the collection of various data items, for example, if we talk about the student entity, then its name, address, course and marks can be grouped together to form the record for the student.

File: A File is a collection of various records of one type of entity

Data Structure

It is a way of organizing or storing data into the memory so that operations on the data can be performed efficiently.

Example are:-

Arrays, Stack, Queue, Tree, Graph etc.

Real life scenario

- Queue at a ticket counter,
- Dictionary
- Queue in printer
- Conveyer belt at the airport
- Stack of plates
- Stack of files

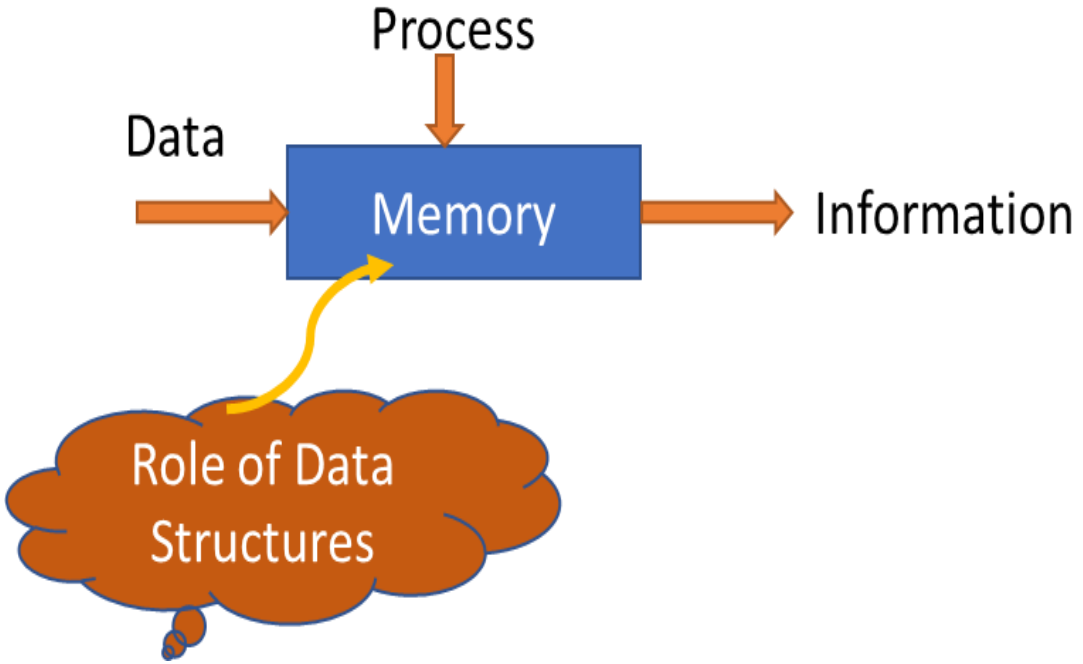
Need for Data Structures

Used to arrange the Data

Helps in management of massive amount of data

Allows Data to be stored in specific manner in Memory

Helps in efficient Data search and Retrieval



Advantages of Data Structures

Efficiency: Efficiency of a program depends upon the choice of data structures.

Reusability: Data structures are reusable, i.e. once we have implemented a particular data structure, we can use it at any other place.

Abstraction: Data structure is specified by the ADT which provides a level of abstraction.

Operation on Data Structures

1. **Create:** create operation results in reserving memory for program elements.
2. **Insertion:** Add an element to the Data Structure
3. **Deletion:** Remove an element from Data Structure, delete first or last element or element at given location.
4. **Traversing:** Visit the element at least once to perform some specific operation like print data element, find sum, average, count nodes in list etc.
5. **Searching:** To find a specific element in given data structure.
6. **Sorting:** arranging elements in a particular order either in ascending or in descending.
7. **Merging:** When two lists List A and List B of size M and N respectively, of similar type of elements, clubbed or joined to produce the third list, List C of size (M+N), then this process is called merging

Abstract Data Type

ADT represents the logical or mathematical model for any data structure.

It basically describes:

a) Representation of data.

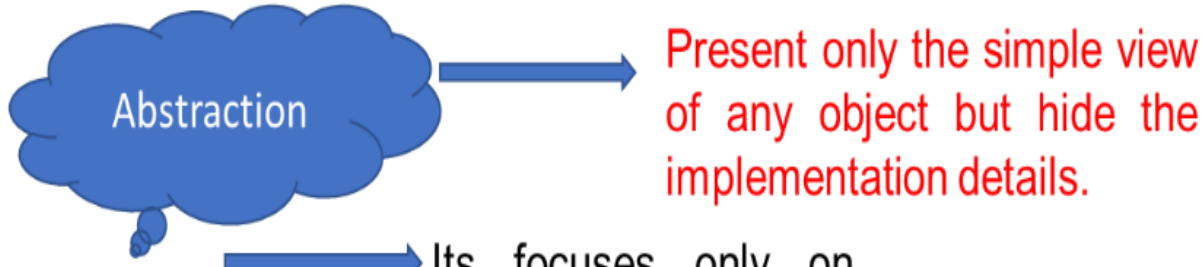
b) Operations that can be performed on these data without any implementation.

An abstract data type strictly is a triple (*D,F,A*) consisting of a set of domains *D*, a set of functions *F* each with range and domain in *D*, and a set of axioms *A*, which specify the properties of the functions in *F*.:

Domain (D): this is the range of values that the data may have
Function(F): this is the set of operations which may legally be applied to elements of data object.
Axioms (A): This is the set of rules with which the different operations belong to F actually be implemented.

Abstract Data Type

- it is a special kind of datatype
- It is defined by a set of values and operations.
- perform different operations, but working of these functions remains hidden from the user.
- Some examples of ADT are List , Stack , Queue etc.



Ex- TV remote

To start the TV ,simply press ON button without knowing its implementation details.

Which of the following is not a primitive operation on the data structure	
A. Selection	B. Creation
D. Deletion	C. All of the above

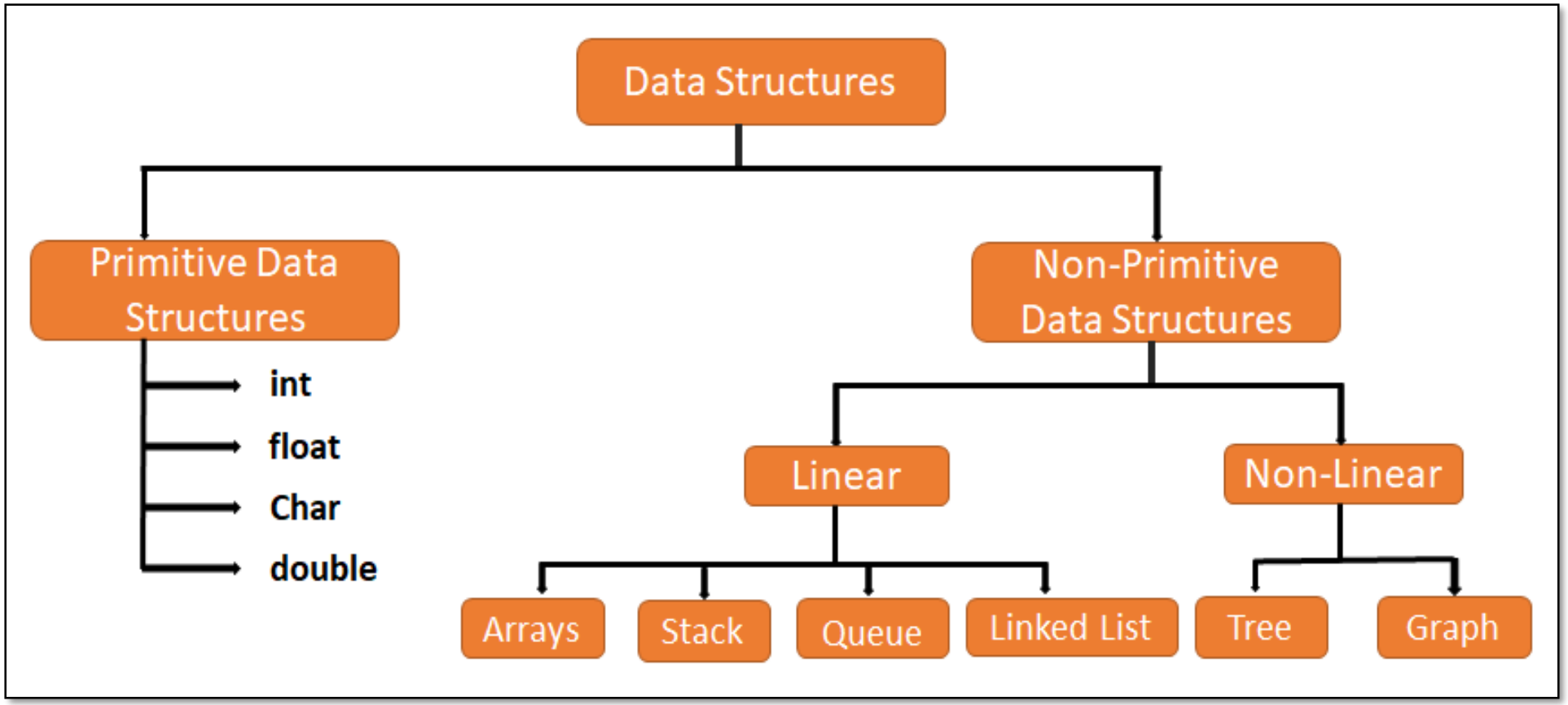
Which of the following is not a primitive operation on the data structure	
A. Selection	B. Creation
D. Deletion	C. All of the above

Types of data structure

Two main types:-

1. Primitive Data Structure
2. Non – Primitive Data Structure

Classification of Data Structures

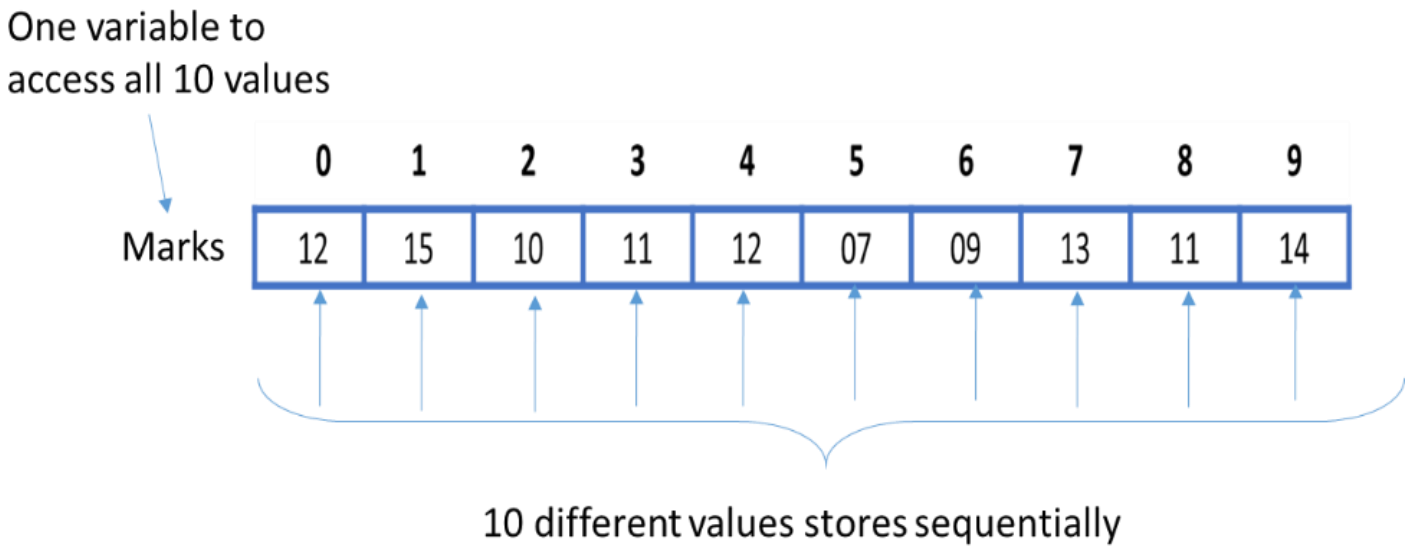


Linear Data Structure

- Type of data structure where arrangement of the data is Linear.
- Examples are
  1. Array
  2. Stack
  3. Linked List
  4. Queue

Array

- It is defined as the collection of similar type of data items stored at contiguous memory locations.



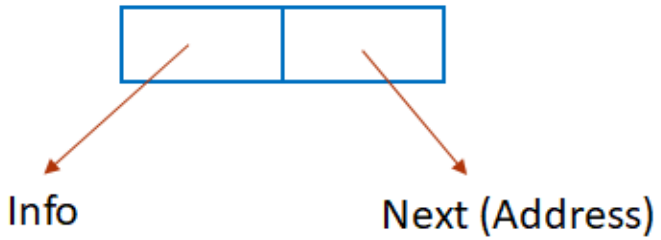
Applications of Array

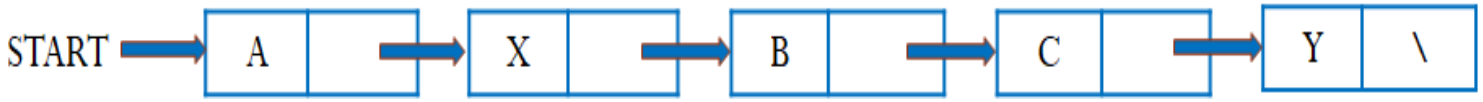
1. The Array data structure is used for memory representation of other linear and non-linear data structures.
2. Matrix or table representation. Database records are maintained by Array.
3. The Array is used for the implementation of various Searching, Hashing and Sorting Algorithms.
4. CPU Scheduling algorithms are also implemented using the Array.

Linear Array is known as	
A. Vertical Array	B. One line Array
D. Straight Line Array	C. One-dimensional

Linked List

- A Linked List is a data structure that consists of nodes.
- Every node contains at least two fields.
- First field contains the information and second field contains address of next node.



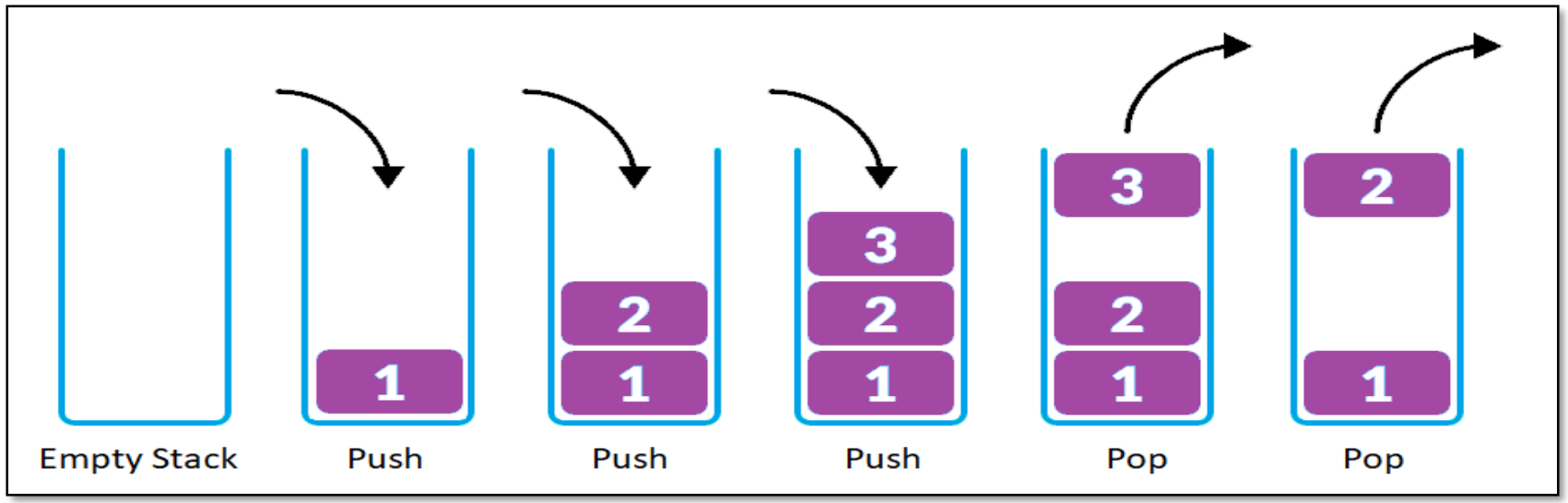


Applications of Link List

1. Memory Organization
2. Web pages (URL Linking)
3. Linux File System: Handling big files using the concept of I-Nodes
4. Implementation of other Data Structures like Stack, Queue, Tree and Graph (Use of Adjacency List).

Stack

- Stack is a data structure where elements can be inserted and deleted from one end, only known as 'Top' of the Stack.
- Insertion in Stack is given a standard name Push and deletion is given a standard name Pop.



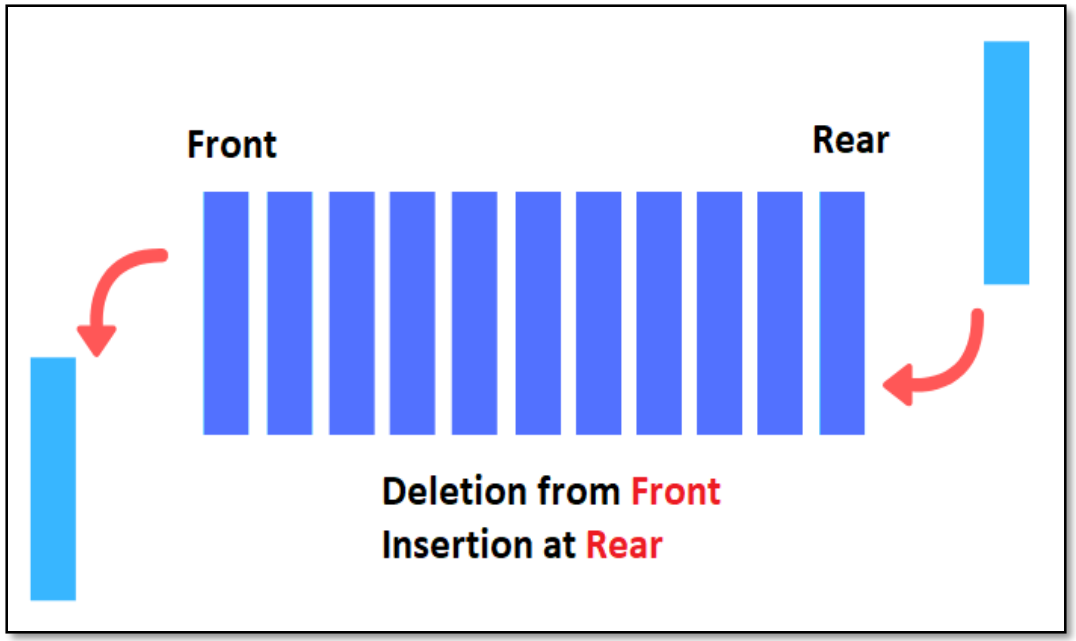
Applications of Stack

1. Stack Data Structure is used for Number Conversion or Expression Evaluation.
2. Reversing of String with the help of Stack.
3. Handling recursion by maintaining Activation Record.
4. Parenthesis Balancing.

A. Which one of the following is an application of Stack Data Structure?	
A. Arithmetic expression evaluation	B. Managing function calls
D. The stock span problem	C All of the above

Queue

- A Queue is an ordered collection of items into which items may be inserted at one end called **Rear** and removed from another end called **Front**.
- Ordered means First in First Out (FIFO) or First Come First Serve (FCFS).



Application Of Queue

1. Queue is used in CPU scheduling Algorithms.
2. Resource allocation management in Operating System.
3. Queue in Network Routing.
4. Interrupt handling in Operating System.

…… form of access is used to add and remove nodes from a queue.	
A. A. LIFO, Last In First Out	B. FIFO, First In First Out
C Both a and b	D. None of these

Removing an element from the stack is …….	
A. Crush	B. Evaluate
D. Pop	C. None of these

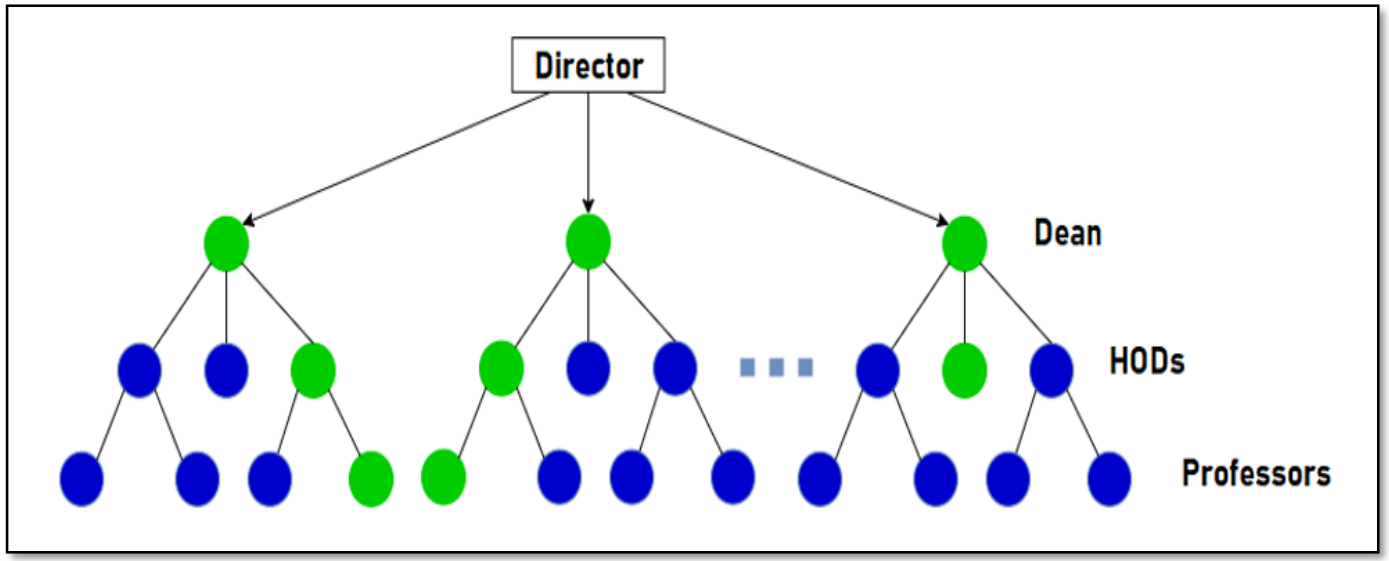
Non - Linear Data Structure

- Data structures where data elements are not arranged sequentially or linearly.
- Examples are

  1. Tree
  2. Graph

Tree

- A Tree is a non-linear hierarchical data structure that follows a Parent-Child relationship. It is a collection of nodes where one node is defined as ROOT node and this root can have zero or more children.

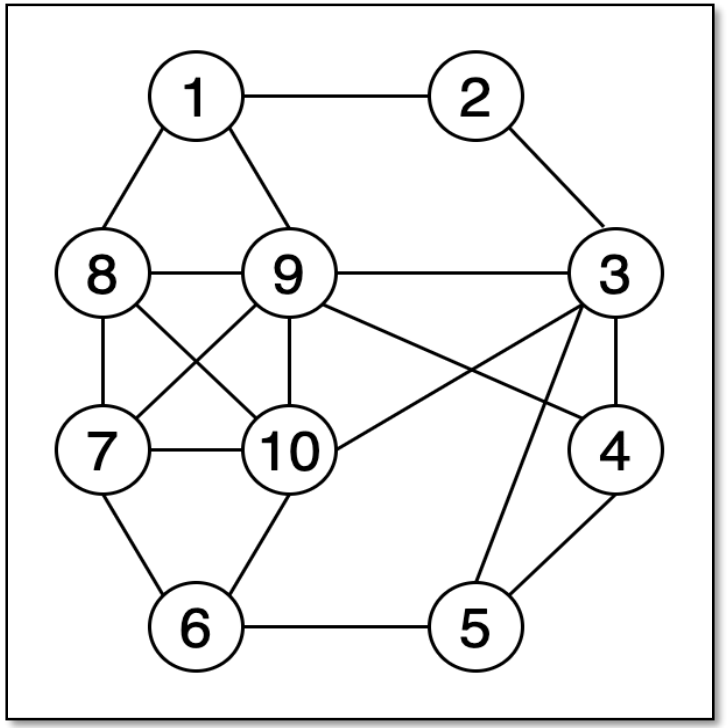


Applications of Tree

1. For compression of text data using Huffman coding
2. Manipulation and storage of hierarchical data.
3. Implementation of indexing in databases with the help of B-Tree and B+-Tree data structure.
4. Representation and evaluation of arithmetic expressions

Graph

- A non-linear data structure consisting of vertices/nodes and edges/lines.
- A Graph can also be termed a mathematical model used to define pair-wise relations between objects.
- A Graph is also defined as an ordered pair G = {V, E}
  where V= a set of vertices and E = a set of edges.



Applications of Graph

1. Google maps is a well-known application of Graph.
2. Social Networks.
3. World Wide Web
4. In Operating System, resource allocation Graph and wait for Graph.
5. In Artificial Neural Networks.

Difference between Linear and Non-Linear Data Structure

Linear Data Structure	Non-Linear Data Structure
Every item is related to its previous and next item.	Every item is attached with many other items.
Data is arranged in linear sequence.	Data is not arranged in sequence.
Data items can be traversed in a single run.	Data cannot be traversed in a single run.
Eg. Array, Stacks, linked list, queue.	Eg. tree, graph.
Implementation is easy.	Implementation is difficult.

A. Pick out the linear data Structure	
A. Tree	B Queue
D. Graph	C. Binary tree

Which of the following data structure non linear?	
A Tree	B. Strings
D. Stack	C. None of the above

Summary

In this lecture we discussed:-

- What is Data Structure
- Importance of Data Structure
- Classification of Data Structure
- Linear and Non-Linear Data Structure
- Application of Data Structure

Thank You