JAIN
DEEMED-TO-BE UNIVERSITY
ONLINE

https://onlinejain.com

Concepts of Data Structures - I

Unit – 5

Prof. Mohan Raj C S

Introduction

Good Computer Program:

- A computer program is a series of instructions to carry out a particular task written in a language that a computer can understand.
- The process of preparing and feeding the instructions into the computer for execution is referred to as programming.
- There are a number of features of a good program
  - Run efficiently and correctly
  - Have a user-friendly interface
  - Be easy to read and understand
  - Be easy to debug
  - Be easy to modify Be easy to maintain

Good Computer Program (Cont…)

- Programs consists of two things: Algorithms and data structures
- A Good Program is a combination of both algorithm and a data structure
- An algorithm is a step by step recipe for solving an instance of a problem
- A data structure represents the logical relationship that exists between individual elements of data to carry out certain tasks
- A data structure defines a way of organizing all data items that consider not only the elements stored but also stores the relationship between the elements

Data Structures

- Data is simply a collection of facts and figures.
- Data is a value or set of values.
- A data item refers to a single unit of values.
- Data items divided into sub-items are known as group items; those **not divided** into sub-items are known as **elementary items**.

For example, a student's name may be divided into three sub-items – [first name, middle name and last name] but a student's ID would normally be treated as a single item.

- A Data structure is a particular way of storing and organizing data in a computer's memory so that it can be used efficiently.

Algorithms

- An algorithm is a step-by-step recipe for solving an instance of a problem.
- Every single procedure that a computer performs is an algorithm.
- An algorithm is a precise procedure for solving a problem in finite number of steps.
- An algorithm states the actions to be executed and the order in which these actions are to be executed.
- An algorithm is a well-ordered collection of clear and simple instructions of definite and effectively computable operations that when executed produce a result and stop executing at some point in a finite amount of time rather than just going on and on infinitely.

Algorithm Properties

- An algorithm possesses the following properties:
  - It must be correct.
  - It must be composed of a series of concrete steps.
  - There can be no ambiguity as to which step will be performed next.
  - It must be composed of a finite number of steps.
  - It must terminate.
  - It takes zero or more inputs
  - It should be efficient and flexible
  - It should use less memory space as much as possible
  - It results in one or more outputs

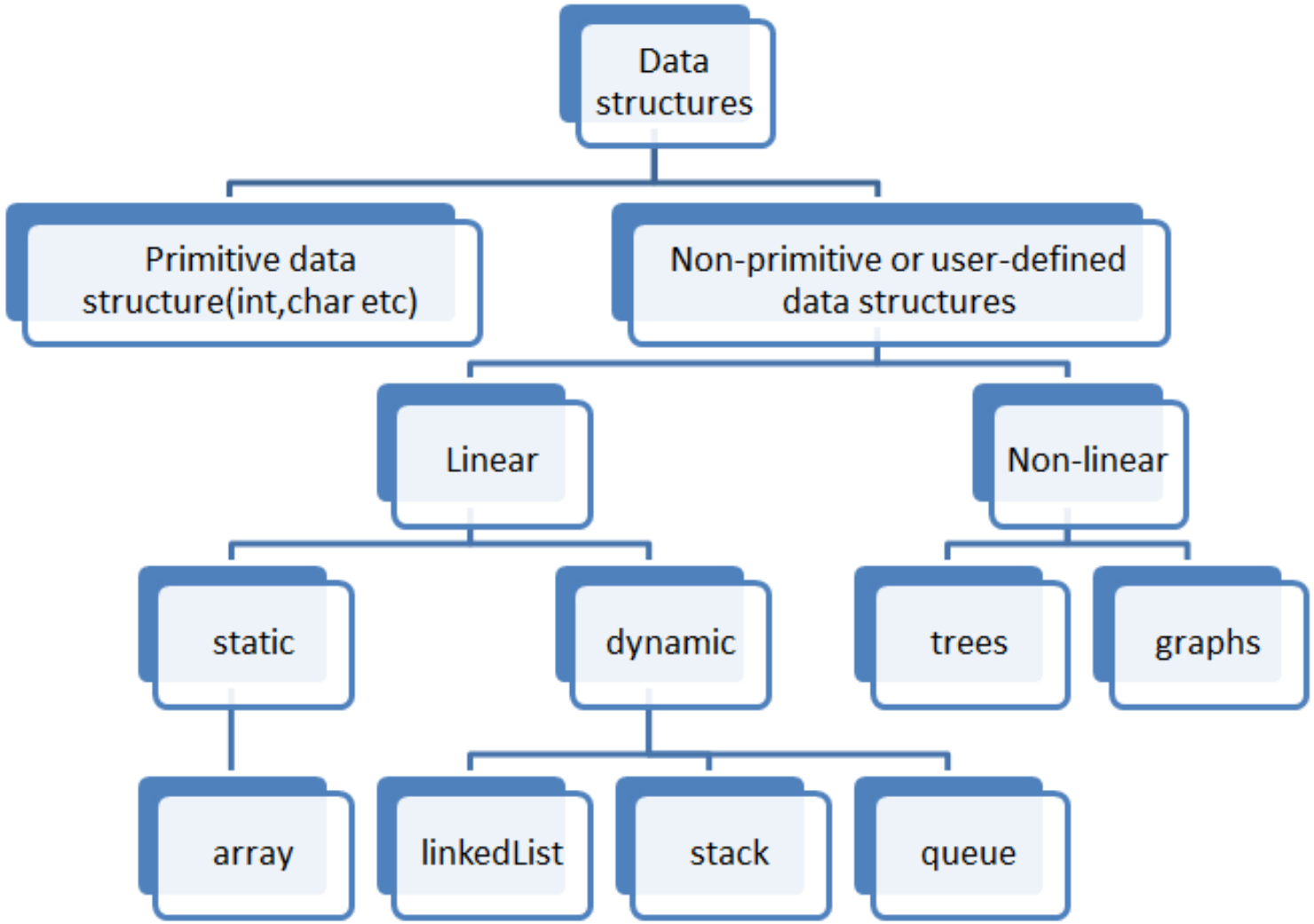
Need for Data Structure

- Data structures organize data
  - ➢ more efficient programs.
  - More powerful computers ➔More complex applications.
  - More complex applications demand more calculations.

Complex computing tasks are unlike our everyday experience.

- More typically, a data structure is meant to be an organization for a collection of data items.
- Any organization for a collection of records can be searched, processed in any order, or modified.
- The choice of data structure and algorithm can make the difference between a program running in a few seconds or many days. A data structure requires a certain amount of:
  - Space for each data item it stores
  - Time to perform a single basic operation
  - Programming effort.

Data Structure Classifications



Primitive Data Structures

- A primitive data structure that is directly operated upon by the machine instructions.
- A primitive data structure can store the value of only one data type. For example, a char data structure (a primitive data structure) can store only characters.

Key features of a primitive data structure:

- The size of a primitive data structure is known as it can store can only one data type.
- The primitive data structure will always store some data. It cannot contain the NULL value.
- Examples of the primitive data type are integer, character, boolean, float, double, long, etc.

Non-Primitive Data Structures

- The non-primitive data structures emphasize the structuring of a group of **homogeneous or heterogeneous** data items.
- **Non-primitive Data Structures:** Unlike the primitive data structure, a non-primitive data structure can store the value of more than one data type.

Key features of a Non - primitive data structure:

- The size of a primitive data structure depends upon the type of data it stores.
- These data structure will be just initialized without storing any data or value. It can contain the NULL value.
- Examples of the non-primitive data types are linked lists, stacks, queues, graphs,trees, etc.

Linear Data Structures & Nonlinear Data Structures

- A data structure is said to be Linear, if its elements are connected in linear fashion by means of logically or in sequence memory locations.
- Nonlinear data structures are those data structure in which data items are not arranged in a sequence.
  - Examples of Non-linear Data Structure are Tree and Graph.

Static Data Structure

- Inefficient as memory is allocated that may not be needed.
- Memory locations allocated will also be sequential which increases the access speed.
- Structures are fixed size which makes them easy and predictable .

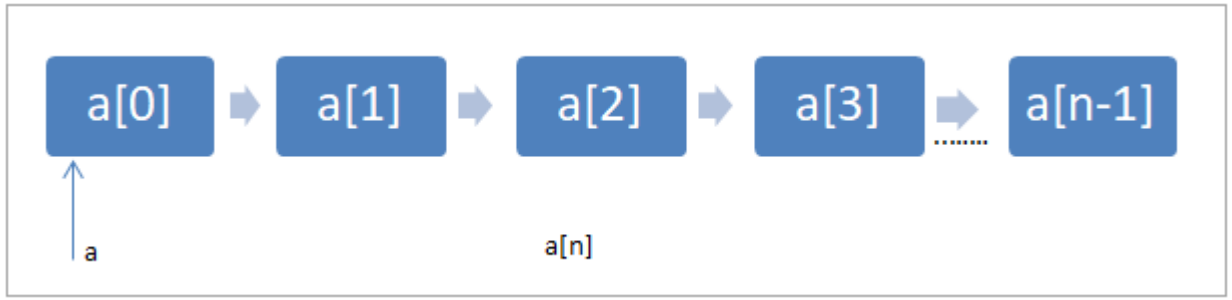
Dynamic Data Structure

- Efficient as the memory varies as needed.
- Memory addresses are usually fragmented which reduces access speed.
- Structure vary in size so it is harder to find the length.

Static Data structure

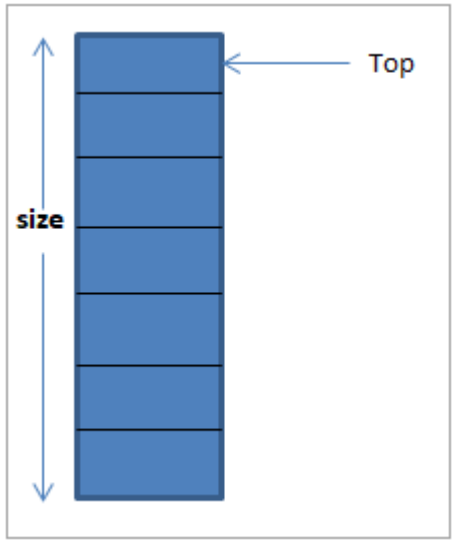
- **Arrays** : An array is a sequential collection of elements of the same type. Each element of the array can be accessed using its position in the array called an index or subscript of the array.

The name of the array points to the first element in the array.

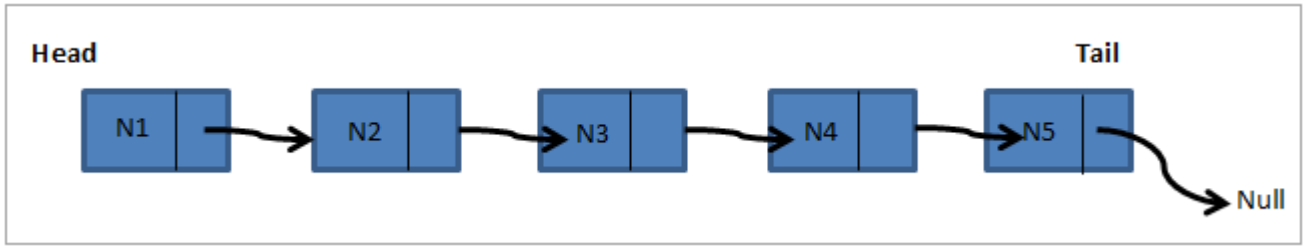


Dynamic Data structures

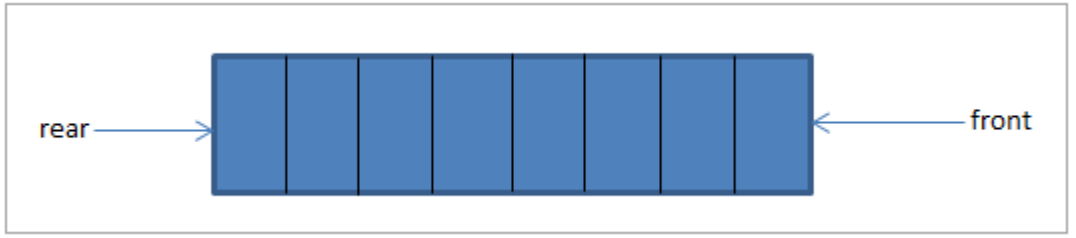
- **Stack**



Linked List



Queues

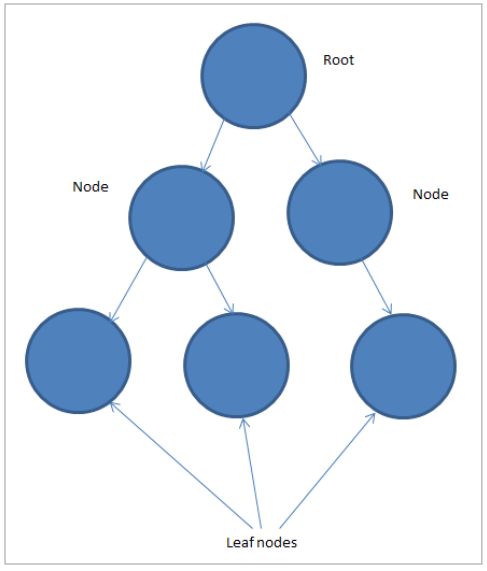


Nonlinear Data Structures

- In non-linear data structures, data is not arranged sequentially, instead, it is arranged in a non-linear fashion.
- Elements are connected to each other in a non-linear arrangement

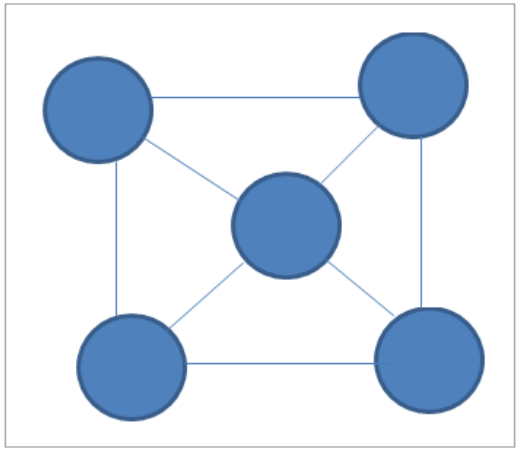
Trees

- Trees are non-linear multilevel data structures having a hierarchical relationship between the elements. Elements of the tree are called Nodes.



Graphs

- The graph is a set of nodes called **vertices** connected to each other by means of the links called **Edges**. Graphs can have a cycle inside it i.e. the same vertex can be a starting point as well as the end point of a particular path. Trees can never have a cycle.



Graph

Operations On Data Structure

These are common to all data structures and are listed as follows:

- Searching
- Sorting
- Insertion
- Deletion
- Traversing

JAIN DEEMED-TO-BE UNIVERSITY | ONLINE

Q & A


THANK YOU