Data Structure-Introduction

Kaustuv Bhattacharjee

University of Engineering & Management, Kolkata

Objective of Good Program

- Runs correctly
- Easy to read and understand (readability)
- Easy to debug
- Easy to modify

Objective of Good Program-Efficiency

- Program should not only give desired results, but also should run *efficiently*

- What is efficient program?

  - Should execute in minimum time
  - Should use minimum memory space
  - Store and retrieve information as fast as possible

Efficiency of a Program-How?

- Application of certain data management concepts

- What are the activities of data management concept?
  - Data collection
  - Data organization (into appropriate structures)
  - Develop and maintain certain routines to achieve quality assurance

Introduction

- Data structure is:
  - a way to organize data in computer so that it can be used efficiently
  - A group of data elements put together under one name, stored and organized in a computer such that can be used efficiently

Usage of Data Structure

- Data structure is widely used in the following areas:
  - Compiler Design
  - DBMS
  - Operating System
  - Graphics
  - Statistical Analysis package
  - Numerical Analysis
  - Artificial Intelligence
  - Simulation

  and so on

Data Structure-Examples

- Arrays
- Linked Lists
- Stacks
- Queues
- Binary Trees
- Hash Tables
- …

Selection of Appropriate Data Structure

- Analyze the objective/performance goal of a given problem
- Figure out the most appropriate data structure for the job

Data Structure-Basic Terminology

- **Data** − Data are values or set of values
- **Data Item** − Data item refers to single unit of values
- **Group Items** − Data items that are divided into sub items are called as Group Items (Ex: Name can be divided into First Name, Middle Name, Last Name)
- **Elementary Items** − Data items that cannot be divided are called as Elementary Items (Ex: Roll Number)
- **Attribute and Entity** − An entity is that which contains certain attributes or properties, which may be assigned values.
- **Entity Set** − Entities of similar attributes form an entity set.
- **Field** − Field is a single elementary unit of information representing an attribute of an entity.
- **Record** − Record is a collection of field values of a given entity.
- **File** − File is a collection of records of the entities in a given entity set.

Data Structure-Basics

Data:

- A value or set of values
- Specifies value of a variable or a constant

Data Definition

- Data Definition defines a particular data with the following characteristics
  - **Atomic** − Definition should define a single concept
  - **Traceable** − Definition should be able to be mapped to some data element
  - **Accurate** − Definition should be unambiguous
  - **Clear and Concise** − Definition should be understandable

Data Structure-Basics Contd…

Data Type (Data Structures)

- Two types:
  - Built-in Data Type (Primary/Language defined /System defined/Primitive/Fundamental)
    - Those data types for which a language has built-in support (Integers, Boolean, Float, Character and Strings)
  - Derived Data Type (User defined/secondary/Non-primitive)
    - Built by the combination of built-in data types (List, Array, Stack, Queue etc)

Data Structure-Classification

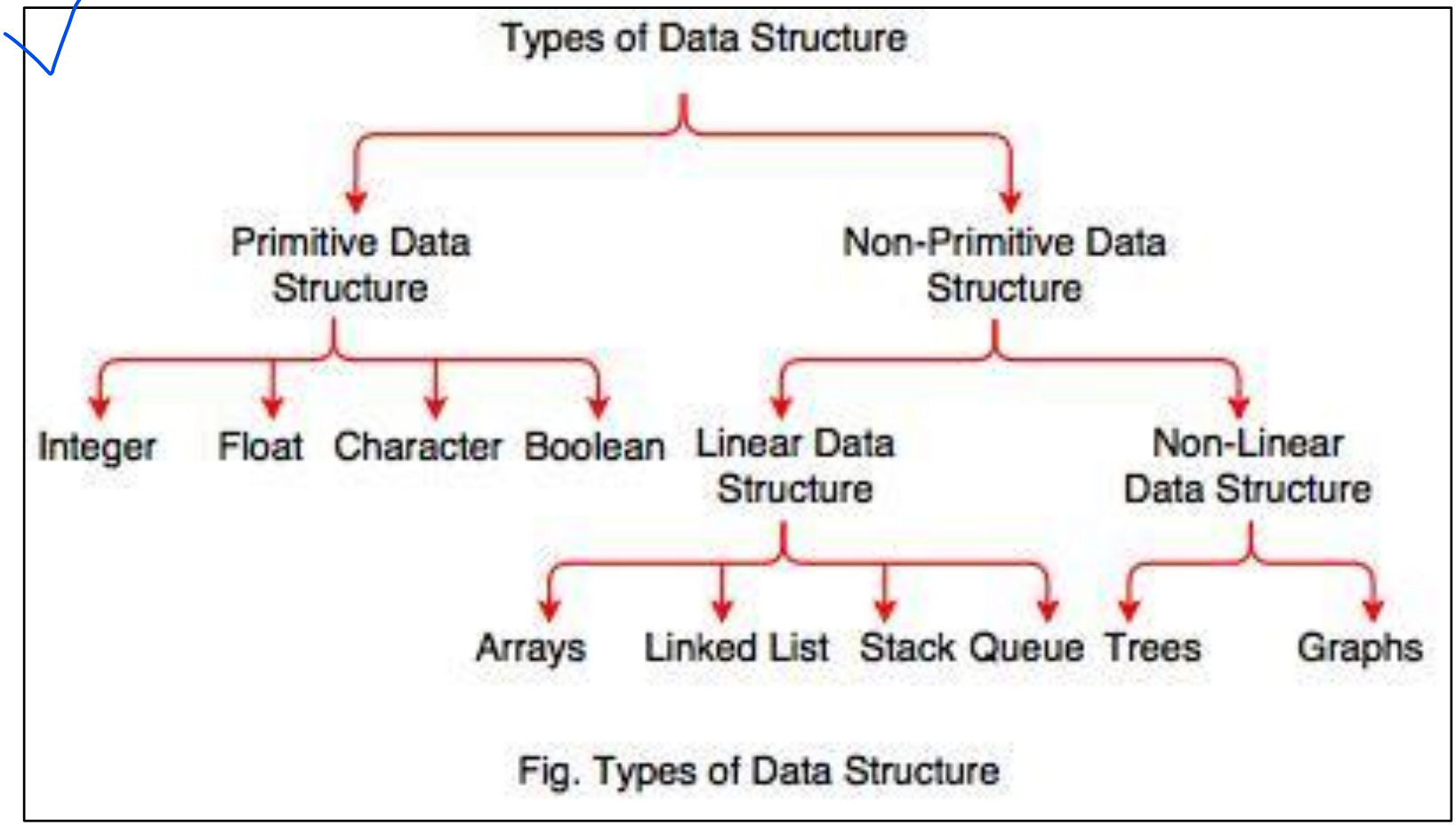


Fig. Types of Data Structure

Linear Data Structure

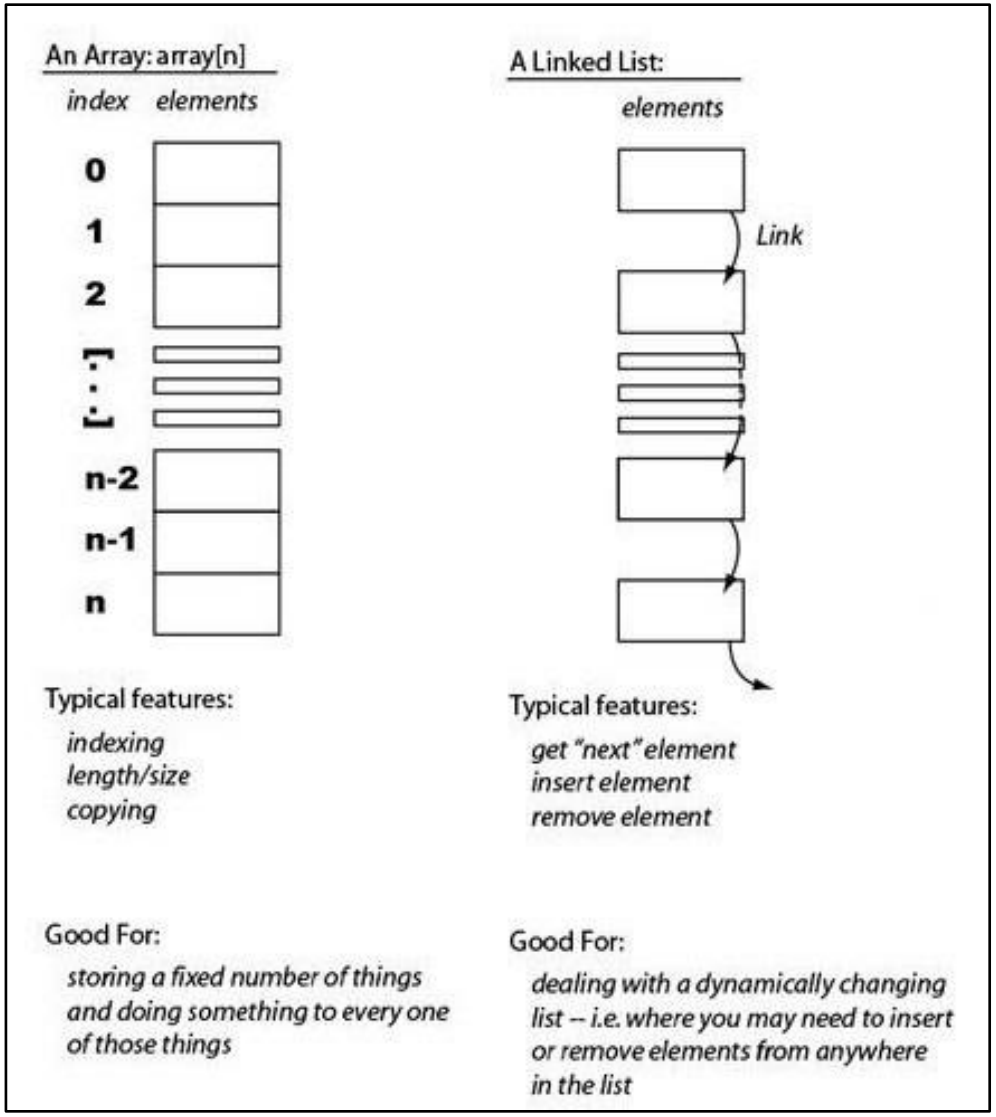
- Data items are organized sequentially or linearly
- Data elements are traversed one after the other and only one element can be directly reached while traversing
- Easy to implement as computer memory is also organized in linear fashion

- **Examples :**
  - Arrays
  - Stack
  - Queue
  - Linked List

Array vs. Linked List

- **Array:**
  - Linear relationship between the elements represented by means of sequential memory locations
  - Useful when number of elements are fixed
- **Linked List:**
  - Linear relationship between the elements represented by pointers
  - Useful when the number of data items are likely to change (dynamic)

Array vs. Linked List-Graphical View



Array vs. Linked List-Comparison Chart

BASIS FOR COMPARISON	ARRAY	LINKED LIST
Basic	It is a consistent set of a fixed number of data items.	It is an ordered set comprising a variable number of data items.
Size	Specified during declaration.	No need to specify; grow and shrink during execution.
Storage Allocation	Element location is allocated during compile time.	Element position is assigned during run time.
Order of the elements	Stored consecutively	Stored randomly
Accessing the element	Direct or randomly accessed, i.e., Specify the array index or subscript.	Sequentially accessed, i.e., Traverse starting from the first node in the list by the pointer.
Insertion and deletion of element	Slow relatively as shifting is required.	Easier, fast and efficient.
Searching	Binary search and linear search	linear search
Memory required	less	More
Memory Utilization	Ineffective	Efficient

Liner Data Structures-Basic Operations

- **Traversing:** Passing through every element one by one which is stored in that particular data structure
- **Insertion:** To insert an element in a particular place of that data structure
- **Deletion:** To delete an element from a particular place of that data structure
- **Searching:** To find the desired element which is stored in that particular data structure
- **Sorting:** To arrange the elements of that particular data structure in ascending / descending order

Non-linear Data Structure

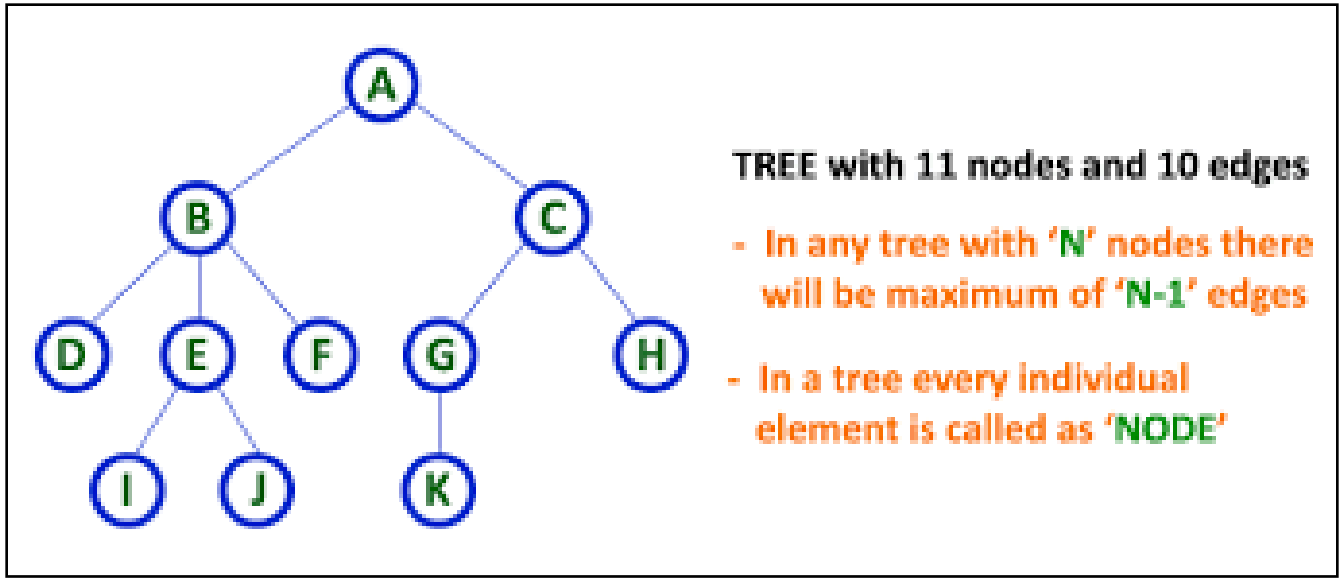
- Data items are not organized sequentially
- A data element could be connected to more than one elements
- All the data elements can not be traversed in single run
- As compared to Linear Data-structure, difficult to implement: Overhead of the link to the next data item
- Efficient memory utilization-free contiguous memory is not required for allocating data items
- The length of the data items is not necessary to be known prior to allocation

- **Examples:**
  - Trees
  - Graphs

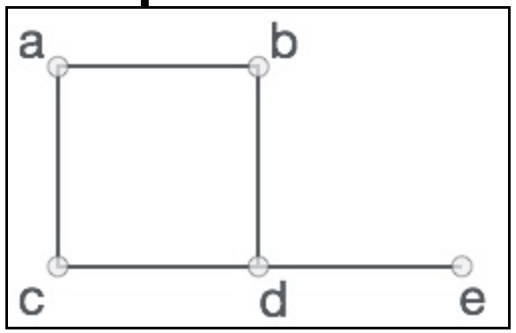
Tree vs. Graph

- **Tree:**
  - Collection of nodes which are arranged hierarchically and form a parent child relationships
  - Access begins from the root node
  - Each node is either a leaf node or an internal node
- **Graph:**
  - A pictorial representation of a set of objects where some pairs of objects are connected by links
  - The interconnected objects are represented by points termed as **vertices**, and the links that connect the vertices are called **edges**.
  - Formally, a graph is a pair of sets (V, E), where V is the set of vertices and E is the set of edges, connecting the pairs of vertices.

Tree vs. Graph-Graphical view



Graph:



Tree vs. Graph

TREE	GRAPH
A data structure that simulates a hierarchical tree structure, with a root value and subtrees of children with a parent node	A data structure that consists of a group of vertices connected through edges
There is a root node	There are no root nodes
There are no loops	There can be loops
Represents data in the form of a tree structure, in a hierarchical manner	Represents data similar to a network
Two major types are binary tree and binary search tree	Two major types directed and undirected graphs
Less complex	More complex

Linear vs Non-linear

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 can not be traversed in a single run.
Examples: Array, Stack, Queue, Linked List.	Examples: Tree, Graph.
Implementation is Easy.	Implementation is Difficult.

Multiple-Choice Questions (MCQ)

1. Which data structure is defined as a collection of similar data elements?

(a) Arrays (b) Linked lists(c) Trees (d) Graphs

2. The data structure used in hierarchical data model is

(a) Array (b) Linked list(c) Tree (d) Graph

3. In a stack, insertion is done at

(a) Top (b) Front(c) Rear (d) Mid

4. The position in a queue from which an element is deleted is called as

(a) Top (b) Front (c) Rear (d) Mid

5. Which data structure has fixed size?

(a) Arrays (b) Linked lists (c) Trees (d) Graphs

6. If TOP = MAX–1, then that the stack is

(a) Empty (b) Full (c) Contains some data (d) None of these

7. Which among the following is a LIFO data structure?

(a) Stacks (b) Linked lists (c) Queues (d) Graphs

8. Which data structure is used to represent complex relationships between the nodes?

(a) Arrays (b) Linked lists (c) Trees (d) Graphs

9. Examples of linear data structures include

(a) Arrays (b) Stacks (c) Queues (d) All of these

Multiple-Choice Questions (MCQ)

Answers:

1. (a) 2. (c) 3. (a) 4. (b) 5. (a) 6. (b) 7. (a), 8. (d) 9. (d)