DATA STRUCTURES AND ALGORITHMS

BY

A.ROGHYA PARVEEN

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & IT

JAMAL MOHAMED COLLEGE (AUTONOMOUS) ,TRICHY.

Introduction to Data Structure

- Computer is an electronic machine which is used for data processing and manipulation.
- When programmer collects such type of data for processing, he would require to store all of them in computer's main memory.
- In order to make computer work we need to know
  - Representation of data in computer.
  - Accessing of data.
  - How to solve problem step by step.
- For doing this task we use data structure

What is Data Structure?

- **A data structure is a particular way of organizing data in a computer so that it can be used effectively**.
- We can also define data structure as a mathematical or logical model of a particular organization of data items.
- Data structure study covers the following points

✓ Amount of memory require to store.

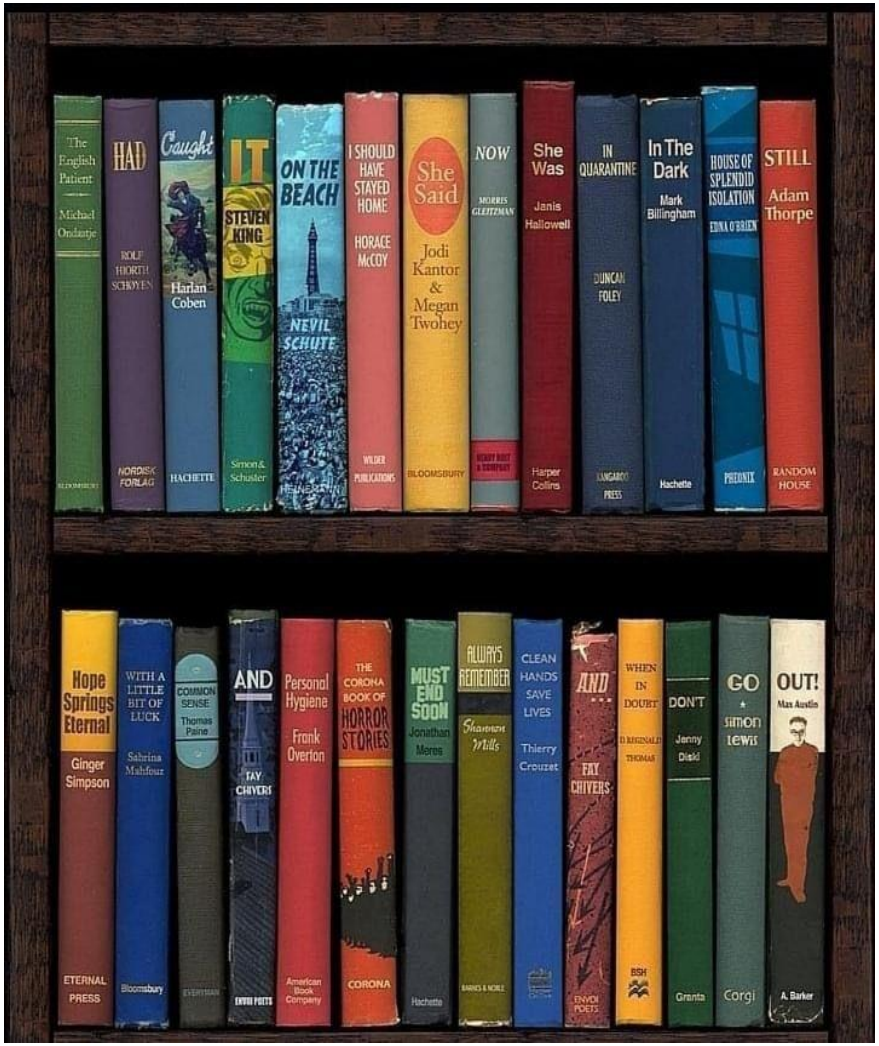
✓ Amount of time require to process.

✓ Representation of data in memory.

✓ Operations performed on that data.

EXAMPLE


alamy stock photo

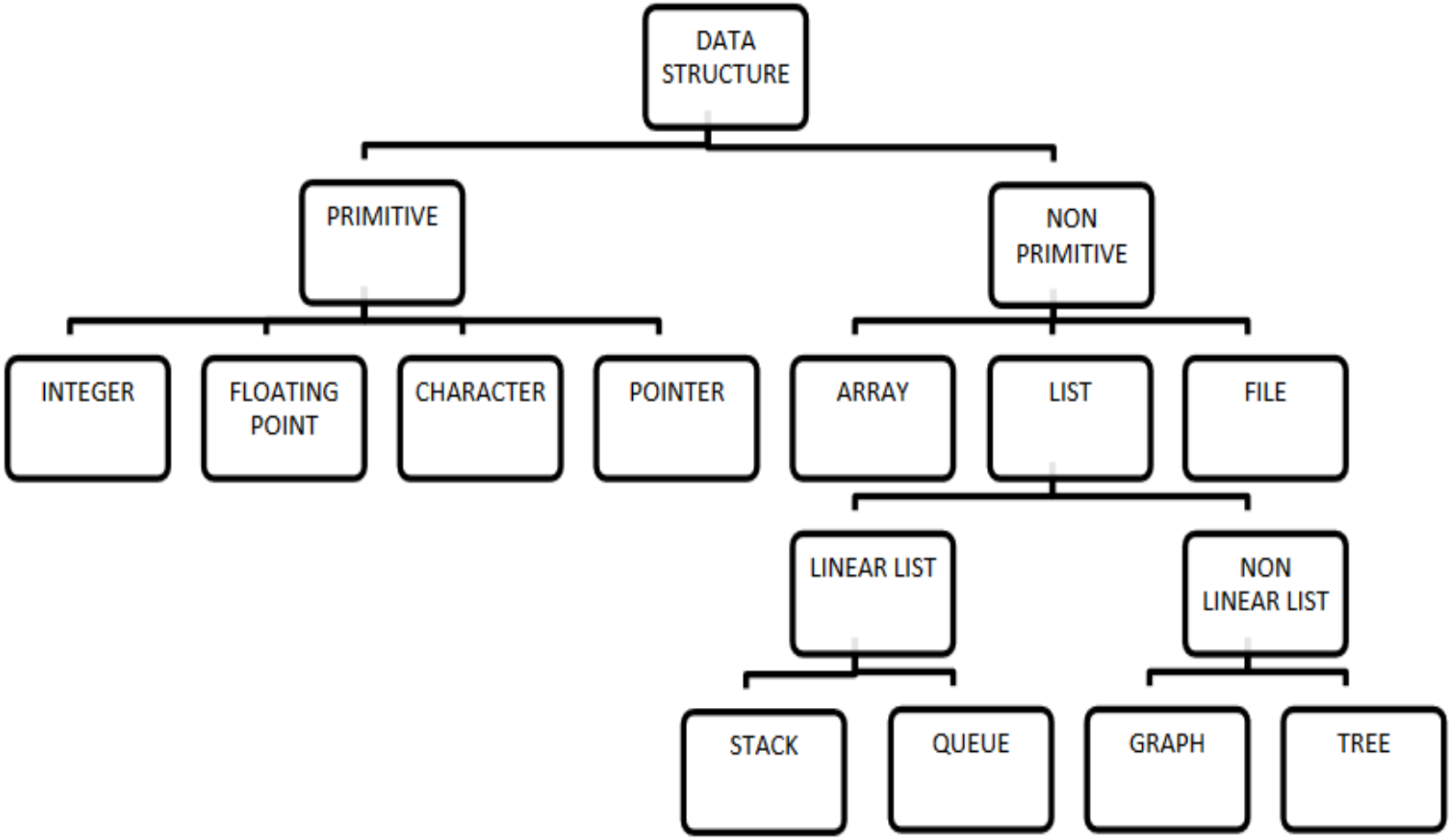

The English Patient
Michael Ondaatje
HAD
ROLF HJORTH SCHOYEN
Caught
Harlan Coben
IT
STEVEN KING
ON THE BEACH
NEVIL SCHUTE
I SHOULD HAVE STAYED HOME
HORACE McCOY
She Said
Jodi Kantor & Megan Twohey
NOW
MORRIS GLEITZMAN
She Was
Janis Hallowell
IN QUARANTINE
DUNCAN FOLEY
In The Dark
Mark Billingham
HOUSE OF SPLENDID ISOLATION
EDNA O'BRIEN
STILL
Adam Thorpe
Hope Springs Eternal
Ginger Simpson
ETERNAL PRESS
WITH A LITTLE BIT OF LUCK
Sabrina Mahfouz
Bloomsbury
COMMON SENSE
Thomas Paine
AND
TAY CHIVERS
Personal Hygiene
Frank Overton
American Book Company
THE CORONA BOOK OF HORROR STORIES
CORONA
MUST END SOON
Jonathan Meres
Hachette
ALWAYS REMEMBER
Shannon Mills
CLEAN HANDS SAVE LIVES
Thierry Crouzet
AND
FAY CHIVERS
ENVOI POETS
WHEN IN DOUBT
BSH
DON'T
Jenny Dield
Granta
GO
SIMON LEWIS
Corgi
OUT!
Max Austin
A. Barker


ENGLISH
DEUTSCH
FRANÇAIS
ITALIANO
ESPAÑOL
PORTUGUÊS


469167628


shutterstock.com · 338992919

Classification of Data Structure



Classification of Data Structure

Data Structures are normally classified into two broad categories

1. Primitive Data Structure
2. Non-primitive data Structure

1.Primitive Data Structure

- Primitive data structures are basic structures and are directly operated upon by machine instructions.
- Primitive data structures have different representations on different computers.
- Integers, floats, character and pointers are examples of primitive data structures.

1.Primitive Data Structure

These data types are available in most programming languages as built in type.

- Integer: It is a data type which allows all values without fraction part. We can use it for whole numbers.
- Float: It is a data type which use for storing fractional numbers.
- Character: It is a data type which is used for character values.
- Pointer: A variable that holds memory address of another variable are called pointer.

2.Non primitive Data Type

- These are more sophisticated data structures.
- These are derived from primitive data structures.
- The non-primitive data structures emphasize on structuring of a group of homogeneous or heterogeneous data items.
- Examples of Non-primitive data type are Array, List, and File etc.

2.Non primitive Data Type

A Non-primitive data type is further divided into Linear and Non-Linear data structure.

- Array:An array is a fixed-size sequenced collection of elements of the same data type.
- List:An ordered set containing variable number of elements is called as Lists.
- File: A file is a collection of logically related information. It can be viewed as a large list of records consisting of various fields.

Linear 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.
- There are two ways to represent a linear data structure in memory.

✓ Static memory allocation

✓ Dynamic memory allocation

- The possible operations on the linear data structure are: Traversal, Insertion, Deletion, Searching, Sorting and Merging.
- Examples of Linear Data Structure are Stack and Queue.

Linear data structures- **Stack**

Stack is a data structure in which insertion and deletion operations are performed at one end only.

- The insertion operation is referred to as ‘PUSH’ and deletion operation is referred to as ‘POP’ operation.
- Stack is also called as Last in First out (LIFO) data structure.

Linear data structures- Queue

The data structure which permits the insertion at one end and Deletion at another end, known as Queue.

- End at which deletion is occurs is known as FRONT end and another end at which insertion occurs is known as REAR end.
- Queue is also called as First in First out (FIFO) data structure

Nonlinear data structures

- 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.

Nonlinear data structures-TREE

- Tree: A tree can be defined as finite set of data items (nodes) in which data items are arranged in branches and sub branches according to requirement.
- Trees represent the hierarchical relationship between various elements.
- Tree consist of nodes connected by edge, the node represented by circle and edge lives connecting to circle.

Nonlinear data structures-GRAPH

Graph is a collection of nodes (Information) and connecting edges (Logical relation) between nodes.

Graphs have many types:

1. Un-directed Graph.
2. Directed Graph.
3. Mixed Graph.
4. Multi Graph.
5. Simple Graph.
6. Null Graph.
7. Weighted Graph

Difference between Linear and Non Linear Data Structure

Linear Data Structure	Non-Linear Data Structure
Every item is related to its previous and next time.	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.

Operation on Data Structures

The most commonly used operations on data structure

1. **Create :** The create operation results in reserving memory for program elements. This can be done by declaration statement. Creation of data structure may take place either during compile-time or run-time. malloc() function of C language is used for creation.
2. **Destroy:** Destroy operation destroys memory space allocated for specified data structure. free() function of C language is used to destroy data structure.
3. **Selection:** Selection operation deals with accessing a particular data within a data structure

Operation on Data Structures

4.**Updation**-It updates or modifies the data in the data structure.

5.**Searching**-It finds the presence of desired data item in the list of data items, it may also find the locations of all elements that satisfy certain conditions.

6.**Sorting**-Sorting is a process of arranging all data items in a data structure in a particular order, say for example, either in ascending order or in descending order.

Operation on Data Structures

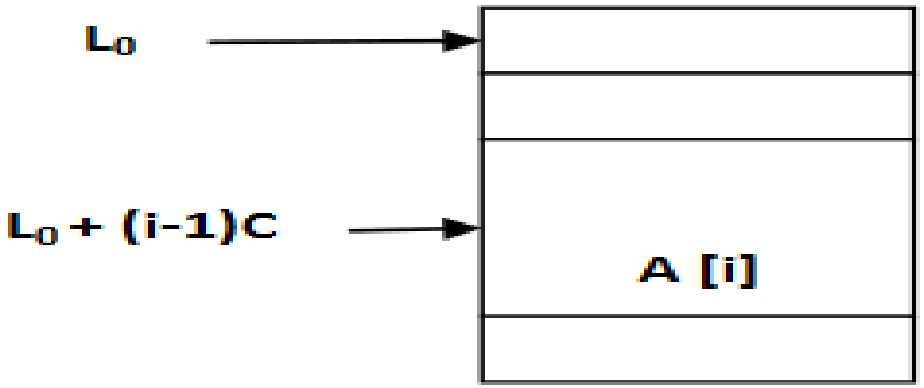
7. **Merging**-Merging is a process of combining the data items of two different sorted list into a single sorted list.
8. **Splitting-** Splitting is a process of partitioning single list to multiple list.
9. **Traversal-**Traversal is a process of visiting each and every node of a list in systematic manner.

Linear Data Structure-ARRAY

One Dimensional Array

- Simplest data structure that makes use of computed address to locate its elements is the one-dimensional array or Vector.
- A vector size is fixed and therefore requires a fixed number of memory locations.
- Vector A with subscript lower bound of “one” is represented as below

Representation of One Dimensional Array



- L0- the address of the first word allocated to the first element of vector A.
- C -words are allocated for each element or node

Representation of One Dimensional Array

- The address of Aiis given equation Loc (Ai) = L0+ C (i-1)
- Let’s consider the more general case of representing a vector A whose lower bound for it’s subscript is given by some variable b.
- The location of Ai is then given by Loc (Ai) = L0+ C (i-b)

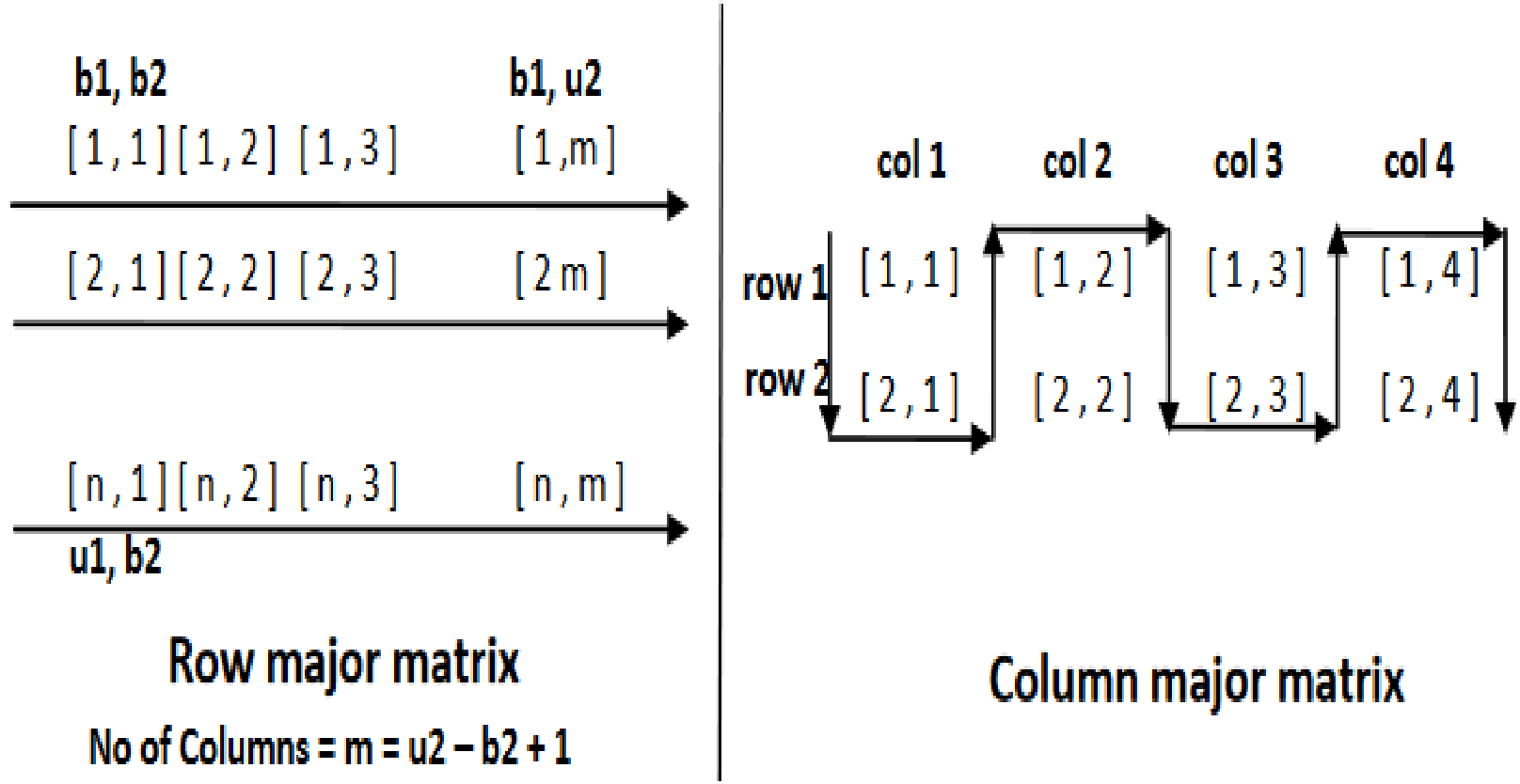
Two Dimensional Array

- Two dimensional arrays are also called table or matrix, two dimensional arrays have two subscripts.
- Two dimensional array in which elements are stored column by column is called as column major matrix.
- Two dimensional array in which elements are stored row by row is called as row major matrix

Two Dimensional Array

- The address of element A [i , j] can be obtained by expression **Loc (A [i , j]) = L0+ (j-1)*2 + i-1**
- In general for two dimensional array consisting of n rows and m columns the address element A [i , j] is given by **Loc (A [i, j]) = L0+ (j-1)*n + (i –1)**
- For row major matrix : **Loc (A [i , j]) = L0+ (i –b1) *(u2-b2+1) + (j-b2)**

Two Dimensional Array



REFERENCES

Books

1. Fundamentals of Data Structure – Ellis Horowiz, SartajSahni, Galgotia

Publications, 2008.

2. Data Structures – Seymour Lipschutz, Tata Mcgraw Hill, Schaum's Outline

Series, 2014.

Websites

1.https://www.learnpick.in/prime/documents/ppts/details/1708/introduction-to-data-structure