Data Structure and Algorithms

1

Deepali Londhe
Information Technology,
PICT, Pune

1

Unit- I Introduction (06 Hrs)

2

- Introduction to Data Structures: Concept of data, Data object, Data structure, Concept of Primitive and non-primitive, linear and Nonlinear, static and dynamic, persistent and ephemeral data structures
- Definition of ADT, Array: Single and multidimensional array address calculation, recursion.
- Searching and sorting: Need of searching and sorting, Concept of internal and external sorting, sort stability
- Searching methods: Linear and binary search algorithms, Fibonacci Series.
- Sorting methods: Bubble, insertion, Quick, Merge, shell and comparison of all sorting methods.
- Case Studies Set Operation, String Operation

2

Unit- I Introduction (06 Hrs)

3

- **Introduction to Data Structures:** Concept of data, Data object, Data structure, Concept of Primitive and non-primitive, linear and Nonlinear, static and dynamic, persistent and ephemeral data structures, Definition of ADT
- **Analysis of algorithm:** Frequency count and its importance in analysis of an algorithm, Time complexity & Space complexity of an algorithm Big 'O', 'Ω' and 'Θ' notations,
- **Sequential Organization:** Single and multidimensional array and address calculation.
- **Linked Organization:** Concept of linked organization, Singly Linked List, Doubly Linked List, Circular Linked List (Operations: Create, Display, Search, Insert, Delete).
- **Case Study** **Set Operation, String Operation**

Contents

4

Section	Contents
DSA Unit-I.1	Introduction to Data Structures, its types
DSA Unit-I.2	Definition of ADT, Array
DSA Unit-I.3	Searching and sorting- Searching
DSA Unit-I.4	**Sorting Methods**

Agenda

5

- Different types of Data Structures
- Primitive and non-primitive Data Structures
- Linear and Nonlinear Data Structures
- Static and dynamic Data Structures
- Persistent and ephemerals

5

Revision

6

- Data
- Data and Information
- Data Object
- What is Data Structure?
- Organization of Data
- Why to use Data Structure?
- Types of Data Structure
- Data structure operations

6

Outcomes

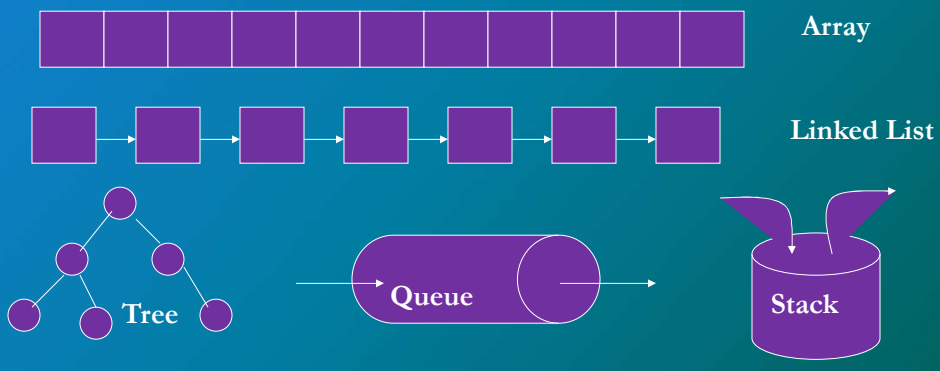
7

- List Different types of Data Structures
- Define and explain with example following data structures
- Primitive and non-primitive Data Structures
- Linear and Nonlinear Data Structures
- Static and dynamic Data Structures
- Persistent and ephemerals

7

Types of data structures

8



There are many, but we named a few. We'll learn these data structures in great detail!

8

Different types of Data Structures

9

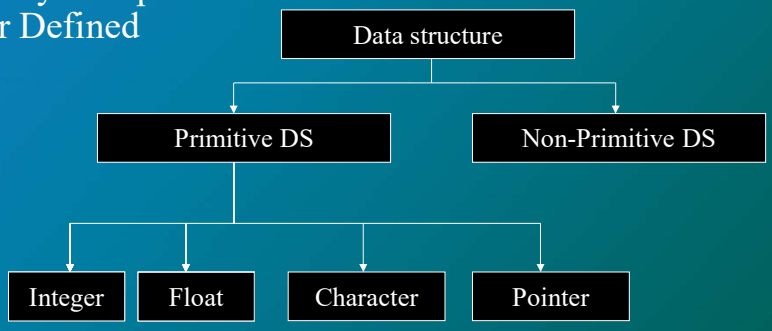
- Primitive and non-primitive
- Linear and Nonlinear
- Static and dynamic
- Persistent and ephemeral

9

Primitive and non-primitive Data Structures

10

Primitive- Predefined by Computer
Non-Primitive – User Defined

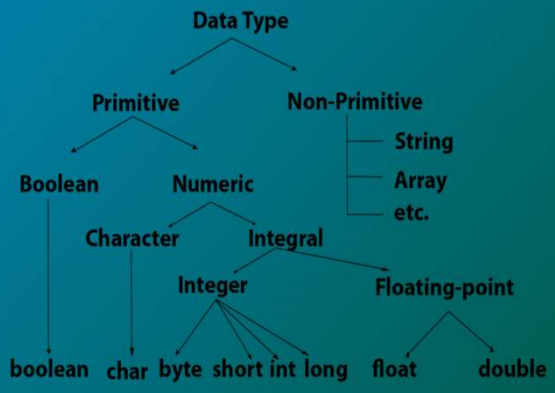


10

Primitive and non-primitive Data Structures

11

- **Primitive data types**
- Predefined types of data, supported by the programming language
- Fits the base architecture of the underlying computer such as int, float, and pointer, and all of the variations.
- Normally is directly operated upon by machine-level instructions



Source: tutsmaster.org

Primitive Data Structures

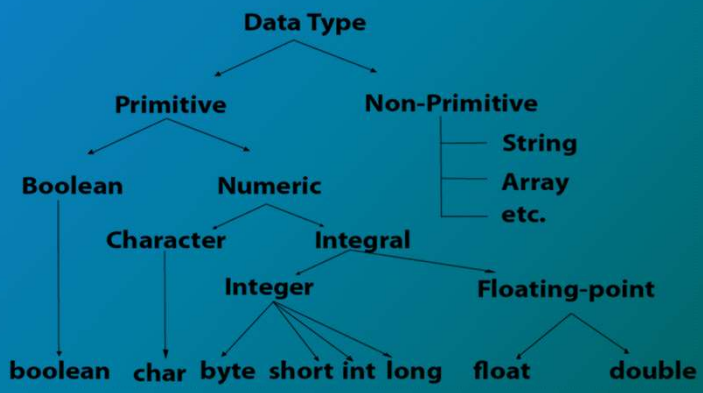
12

- **Integer**
  - stored using a word of memory, 4 bytes or 32 bits,
  - integers from 0 up to 4,294,967,295 (2^{32} - 1) can be stored
  - 00000001 00000000 00000000 00000000 | 1
  - 00000010 00000000 00000000 00000000 | 2
  - 00000101 00000000 00000000 00000000 | 5
- **Character/Text**
  - **Convert character into number, ASCII encoding**
  - **ASCII code for A to Z is 65 to 90**
  - A letter is usually stored using a single byte (8 bits)
  - *"hello"* (104, 101, 108, 108, 111)
  - 01101000 01100101 01101100 01101100 01101111

Assignment-1.1

13

Identify How Real Numbers/floating Point numbers are represented in memory.

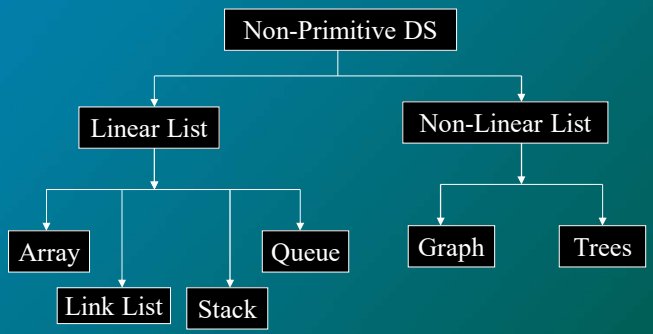


13

Non-primitive Data Structures

14

- More sophisticated data structures.
- Derived from the primitive data structures.
- Emphasize on structuring of a group of homogeneous (same type) or heterogeneous (different type) data items.



14

Primitive and non-primitive Data Structures Contd..

15

- **Non-primitive data types**
- Not defined by the programming language, but are instead created by the programmer.
- Derived from primary data types.
- Store the group of values.
- e.g.Array, structure, union, link list, stacks, queue etc…

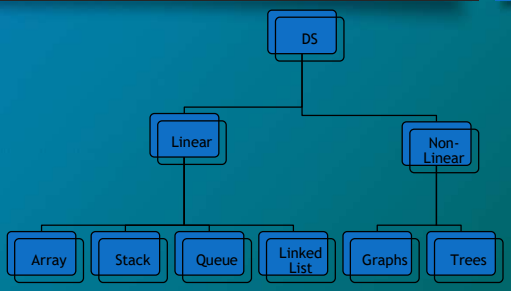
- int a[10]; 4 bytes each

addr	data
1000	a[0]
1004	a[1]
1008	a[2]
1012	a[3]
…..	…
1036	a[9]

Linear and Nonlinear data Structures

16

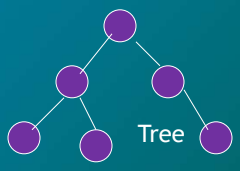
- **Linear Data structure**
- Data elements are arranged sequentially or linearly.
- Single level
- Traverse all the elements in single run only.
- Easy to implement because computer memory is arranged in a linear way.

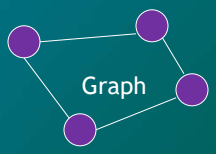


Linear and Nonlinear data Structures

17

- **Non-Linear Data structure**
- Data elements are not arranged sequentially or linearly.
- Single level is not involved
- Can't traverse all the elements in single run
- Not easy to implement.
- Efficient Memory utilization





17

Static and dynamic Data Structure

18

- Static data structure
- Size of the structure is fixed.
- The content of the data structure can be modified but without changing the memory space allocated to it.
- E.g. Array(int a[10])

addr	data
1000	a[0]
1004	a[1]
1008	a[2]
1012	a[3]
…..	…
1036	a[9]

Array

18

Static and dynamic Data Structure

19

- Dynamic data structure
- Size of the structure in not fixed, can be modified during the operations performed on it.
- Dynamic data structures are designed to facilitate change of data structures in the run time.

Linked List

Persistent and ephemeral Data Structures

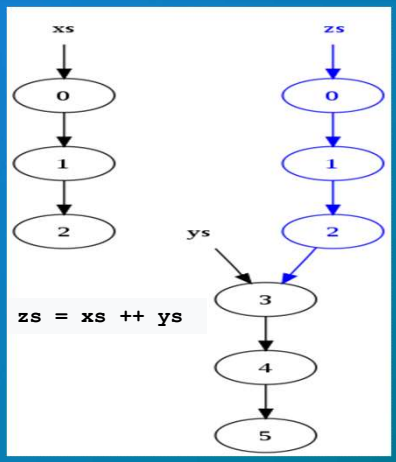
20

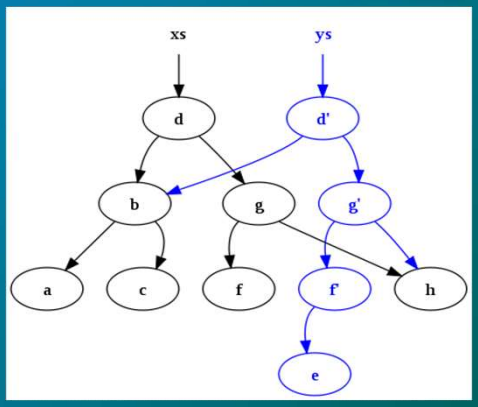
- **Persistent data structure** is a **data structure that always preserves** the previous version of itself when it is modified.
- Such data structures are effectively immutable, as their operations do not (visibly) update the structure in-place, but instead always yield a new updated structure.
- **Partially persistent** DS -if all versions can be accessed but only the newest version can be modified.
- **Fully persistent DS-** if every version can be both accessed and modified.
- If there is also a meld or merge operation that can create a new version from two previous versions, the data structure is called **confluently persistent**.
- Data Structures that are not persistent are called *ephemeral*.

Source: Wikipedia

Persistent and ephemeral Data Structures..

21





References

22

- Books
- R. Gilberg, B. Forouzan, "Data Structures: A pseudo Code Approach with C++", Cengage Learning, ISBN 9788131503140.
- E. Horowitz, S. Sahni, D. Mehta, "Fundamentals of Data Structures in C++", Galgotia Book Source, New Delhi, 1995, ISBN 16782928
- Dinesh P. Shah, Sartaj Sahani , “Handbook of DATA STRUCTURES and APPLICATIONS”, CHAPMAN & HALL/CRC

- Web
  - http://statmath.wu.ac.at/courses/data-analysis/itdtHTML/node55.html
  - https://en.wikipedia.org/wiki/Persistent_data_structure

THANK YOU !