| **1** |  | **Introduction to Data Structures** | 2 |
| --- | --- | --- | --- |
|  | 1.1 | Introduction to Data Structures, Concept of ADT, Types of Data Structures-  Linear and Nonlinear, Operations on Data Structures. |  |
|  |  |  |  |
| **4** |  | **Trees** | 11 |
|  |  | 4.1 Introduction, Tree Terminologies, Binary Tree, Binary Tree Representation, Types of Binary Tree, Binary Tree Traversals, Binary Search Tree, Operations on Binary Search Tree, Applications of Binary Tree-Expression Tree, Huffman Encoding, Search Trees-AVL, rotations in AVL Tree, operations on AVL Tree,  Introduction of B Tree, B+ Tree. |  |
| **5** |  | **Graphs** | 4 |

|  | 5.1 | Introduction, Graph Terminologies, Representation of Graph, Graph Traversals Depth First Search (DFS) and Breadth-First Search (BFS), Graph Application Topological Sorting. |  |
| --- | --- | --- | --- |
| **6** |  | **Searching Techniques** | 4 |
|  | 6.1 | Linear Search, Binary Search, Hashing-Concept, Hash Functions, Collision resolution Techniques |  |

***Data***

The quantities, characters, or symbols on which operations are performed on a computer are Data.

HSAY SI YM EMAN. [ DATA ] YASH IS MY NAME. [ INFORMATION ]

* If data is arranged in a systematic way, then it gets a structure and becomes meaningful. Such meaningful or processed data is called Information.

--------------------------------------------------------------------------------------------

To provide an appropriate way to structure the data, we learn DATA STRUCTURES.

--------------------------------------------------------------------------------------------

***DATA TYPES***

* Defines a certain domain of values.
* Defines the operations allowed on those values.

***User-Defined Data Types***

* Values and Operations are defined by the user. Made by combining other existing data types.
* Example. Structure, Union, and Enumeration.

--------------------------------------------------------------------------------------------

***Abstract Data Types*** *( provides Abstraction )*

ADT is a concept of Data Structure, Data Structures are a realization of ADT.

* ADT - D-ata   
   F-unctions   
   A-xioms
* It’s the BluePrint, which defines operations on values using functions.  
  { Without specifying what’s inside the functions and how the operations are performed. }
* Example. STACK ADT  
  A stack consists of elements of the same type arranged in sequential order.  
  Operations:  
  1. Initialize() - Initialize an empty stack   
  2. IsEmpty() - Checks if the stack is empty  
  3. Push() - Insert an element into the stack  
  4. Pop() - Remove and return an element into the stack  
  5. IsFull() - Checks if the stack is full
* Multiple ways to implement ADTs  
  Example - STACK ADTs are implemented using Arrays and Linked Lists.
* Uses - In a final program/application, the consumer can only access the   
  ADT (interface).

--------------------------------------------------------------------------------------------

***Data Structures***

* A data structure is a systematic way to structure/store and organizes data so that it can be used efficiently.
* Used to implement an ADT.
* ADT tells us what is to be done and DS tells us how to do it.

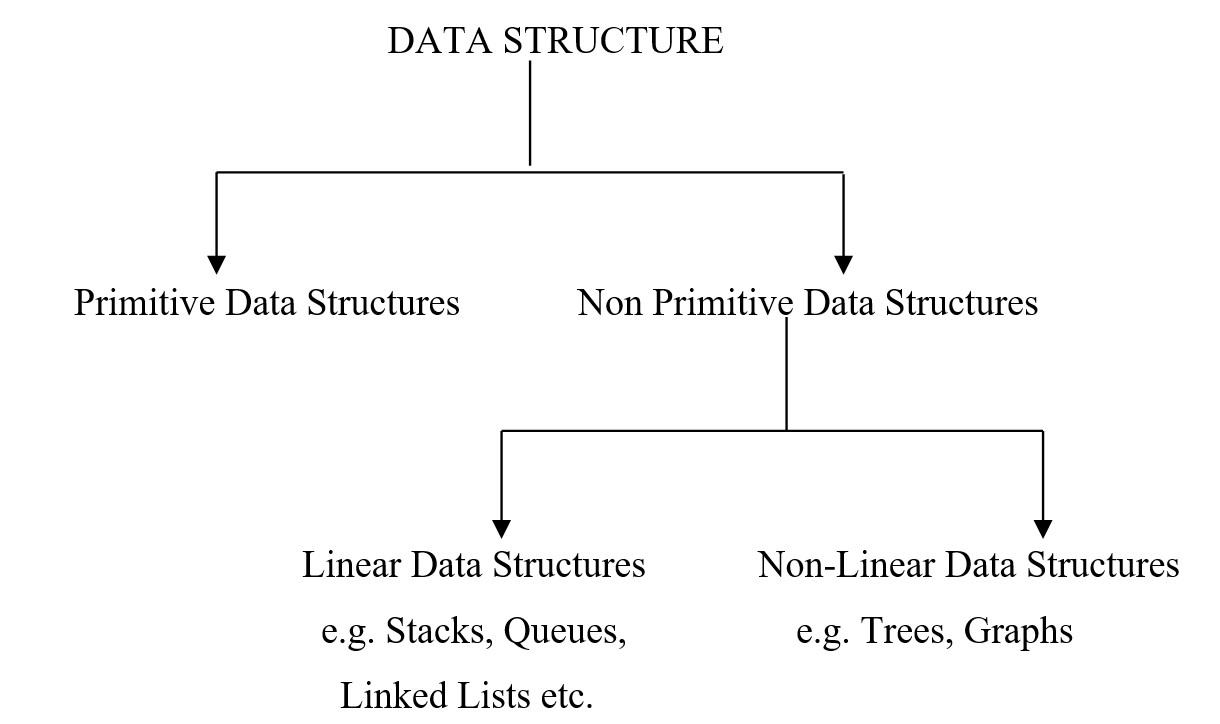
1. Primitive = Fundamental Data types supported by the programming language.

(Simple, Atomic, Built-in, System Defined, Hardware Implemented, Predefined, Basic)

Eg. int, float, bool, pointer, etc.

2. Non Primitive = User-defined/derived using primitive data structures.

Eg. Array, Structure, Stack, etc



Subdivided into 2 types:  
1. Linear (Array, Linked List) There is one to one relationship between the elements.

Every data element has a unique successor and predecessor  
 v/s

Non-Linear (Trees, Graphs)

Every data element may have more than one predecessor as well as a successor.

2. Static (Array)

Has a fixed memory size and the memory is allocated during compilation.

v/s

Dynamic (Linked List)

Does not have a fixed size and the memory can be allocated during runtime.

--------------------------------------------------------------------------------------------

OPERATIONS ON DATA STRUCTURES:

1. Traversing: It means to access each data item exactly once so that it

can be processed.

2. Searching: It is used to find the location of one or more data items that

satisfy the given constraint.

3. Inserting: It is used to add new data items to the given list of data items.

4. Deleting: It means to delete a particular data item from the given

collection of data items.

5. Sorting: Data items can be arranged in some order like ascending order

or descending order depending on the type of application.

6. Merging: Lists of two sorted data items can be combined to form a

a single list of sorted data items.

--------------------------------------------------------------------------------------------

Time Complexity - the amount of time taken by an algorithm to run as a function

Space Complexity - the amount of space/memory taken by the algorithm to run as a function.

We use Big O notation to determine the performance of the algorithm.

--------------------------------------------------------------------------------------------

Identifier vs Variable vs Mnemonic

--------------------------------------------------------------------------------------------

An approximate measure of time is complexity is known as Asymptotic Complexity.

Big O Notation

We use Big O notation to determine the performance of the algorithm.

