

## What is Data structure?

→ Data can be arranged in a many ways, logical or mathematical arrangement of a data is called Data structure.

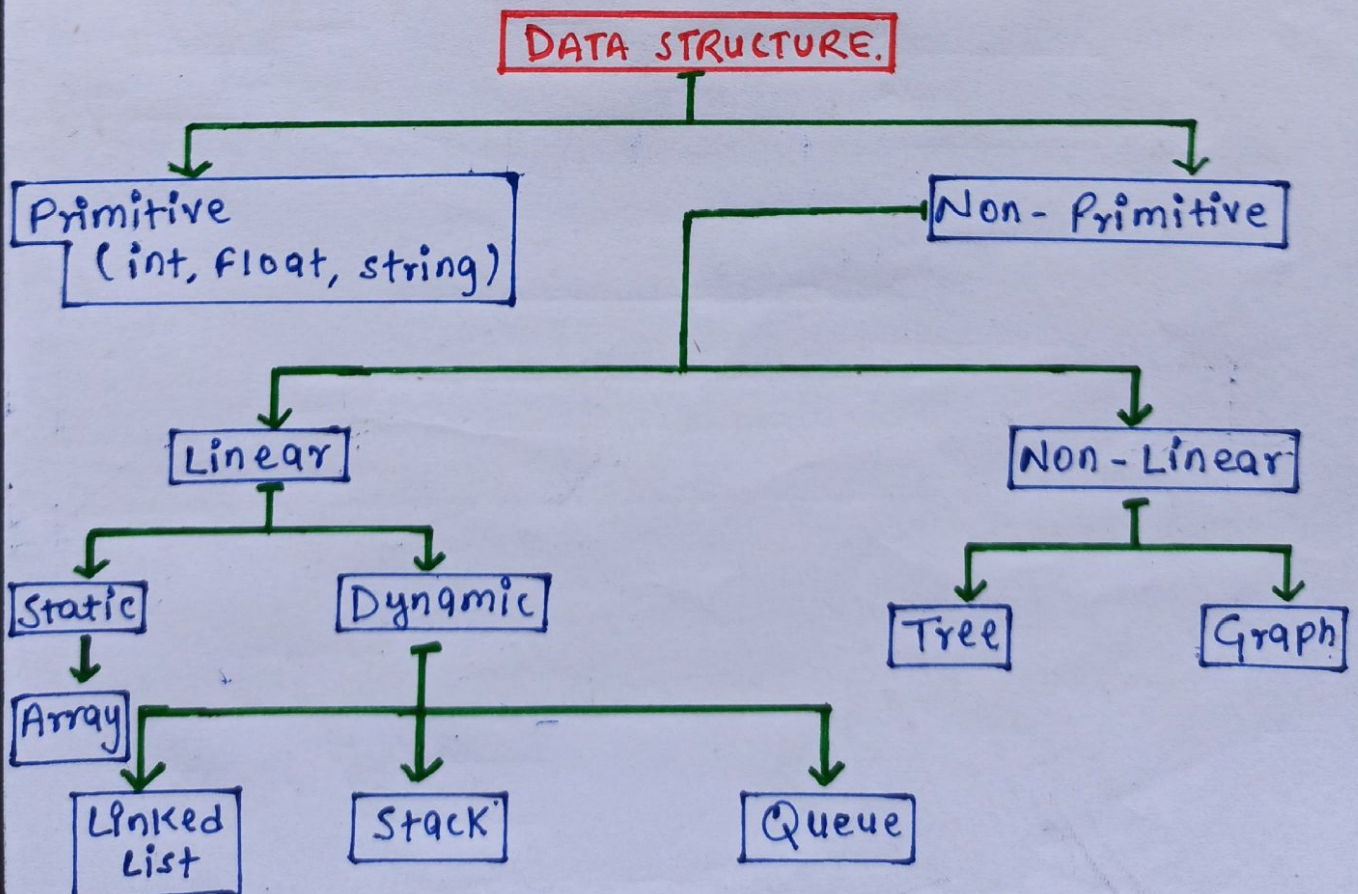
Examples: Array, linkedlist, stack, Queue, Tree Graph and many more.

## What is an Algorithms?

→ Sequence of steps performed on the data using efficient data structures to solve a given problem.

Example: Sorting an Array.

## CLASSIFICATION OF DATA STRUCTURE





## Types of Data Structures

- a Primitive and non-Primitive Data structure.
- b Static and Dynamic Data structure.
- c Persistent and ephemeral Data structure.

Non-Primitive further Divided into two Types.

- i). Linear Data Structure.
- ii). Non-Linear Data Structure.

Persistent further Divided into three types.

- i). Partially Persistent.
- ii). Fully Persistent.
- iii). Confluently Persistent.

## Data Structure Operations:

The following four operations play a major role.

- (1). **Traversing :** Accessing each record exactly once so that certain items in the record may be processed.
- (2). **Searching :** Finding the location of the record with a given key value.
- (3). **Inserting :** Adding a new record to the structure.
- (4). **Deleting :** Removing a record from the structure.
- (5). **Merging :** Combining the records in two different sorted files into a single sorted file.
- (6). **Sorting :** Arranging the record in some logical order  
Example :- Alphabetically according to some NAME key or in Numerical order according to some NUMBER key.



## Searching Algorithms :

A search algorithm is a step-by-step procedure used to locate specific data among a collection of data.

### Types of search algorithms with the complexity

#### 1). Linear Search :

A linear search or sequential search is a method for finding an element within a list. It sequentially checks each element of the list until a match is found or the whole list has been searched.

$$C(n) = n/2$$

← Complexity of linear search.

#### 2). Binary Search :

In binary search approach the element is always searched in the middle of a portion of an array.

Binary search can be implemented only on a sorted list of items.

If the elements are not sorted already, we need to sort them first.

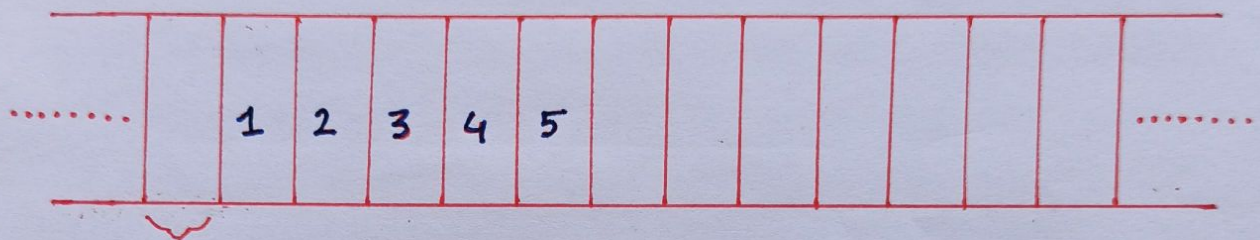
$$C(n) = \log_2 n$$

← Complexity of Binary search.



## ARRAY :

Array is a Type of linear Data structure (OR)  
Array is a collection of more than one data but all the data items are same data types, & stored that data in a computer in a contiguous memory location.



Memory is a long top of Bytes.

## Types of Array :

### ① One Dimensional Array :

The array with only subscript that array is called as **One Dimensional Array**.

Example: `int a[5];` ← Subscript.

### ② Two Dimensional Array :

The array with two subscript that array is called as **Two Dimensional Array**.

Example: `int a[5][5];` ← Subscript.

### ③ Multi-Dimensional Array :

The array with more than two subscript that array is called as **Multi-Dimensional Array**.



## LINKED LIST :-

Linked list is a linear Data structure. It is also a collection of more than one data items of a dissimilar data type like array but it can not store it in contiguous memory location. It can be stored randomly in a main memory.

So that linked list contains two parts: one for Data and second part for the Address of the next data element.

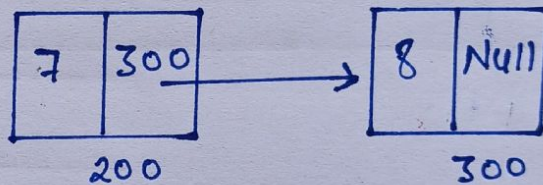
Data element



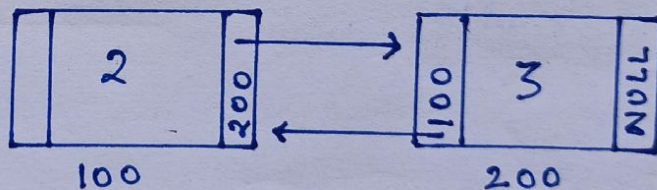
Address

### Types of linked list

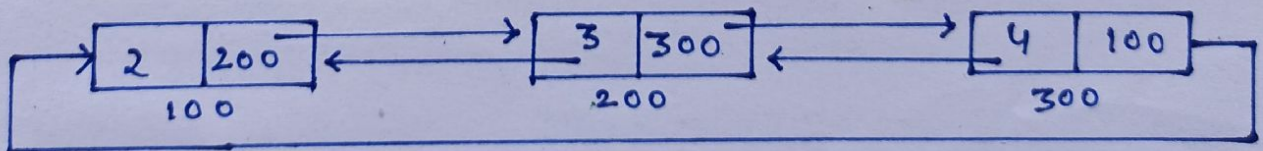
① Singly linked list :



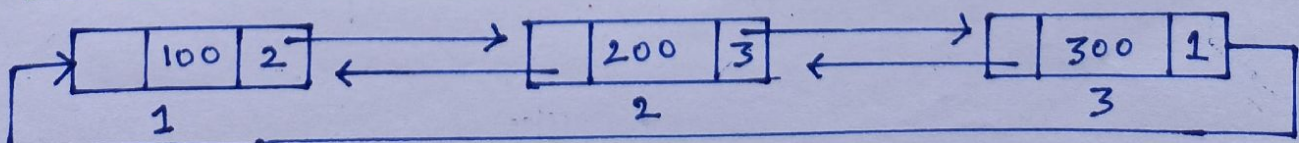
② Doubly linked list :



③ Circular linked list :



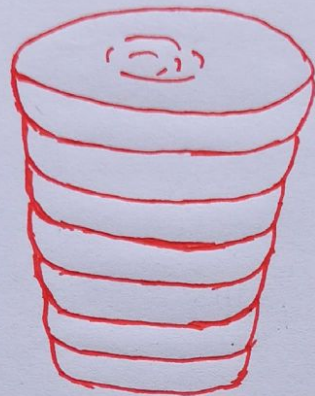
④ Doubly circular linked list :





## STACKS

A stack is a list of elements in which an elements may be inserted or deleted only at one end called the Top of the stack.

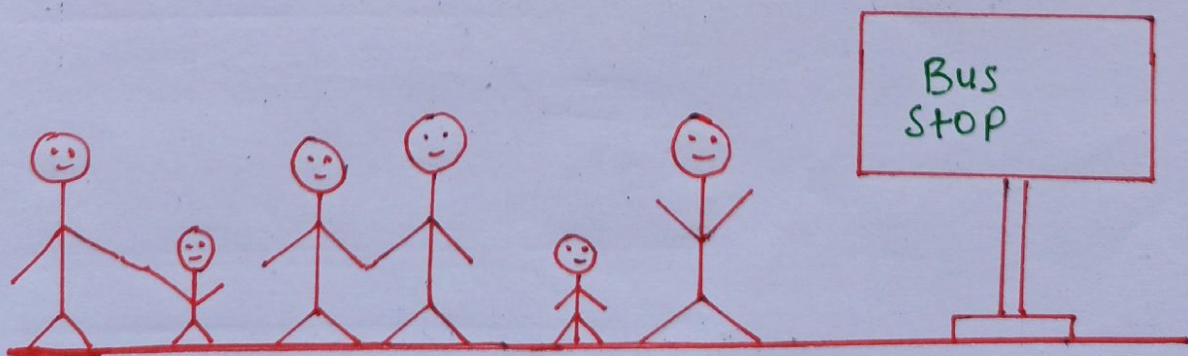


Stack of dishes

Push → Insert elements into stack  
Delete elements from stack ← Pop

## QUEUES

A Queue is a linear list of elements in which deletions can take place only at one end called front and insertions can take place only at the other end called the rear.

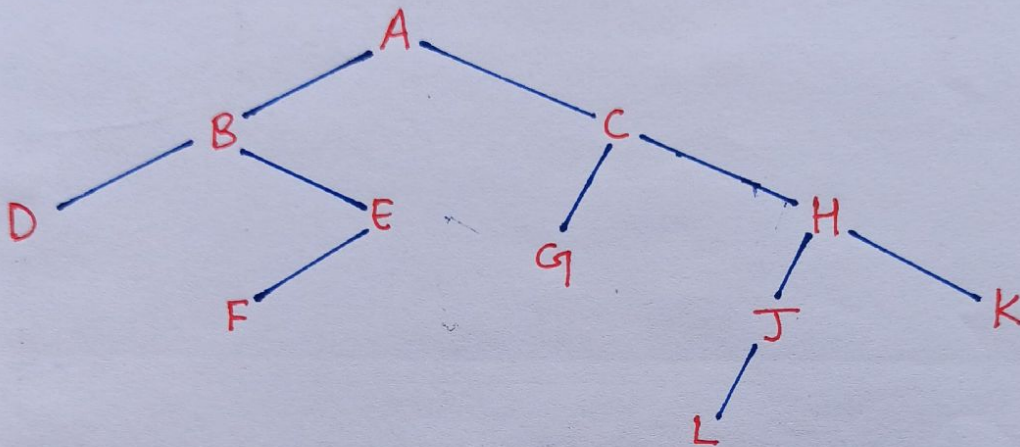




# TREES

**Trees** are non-linear data structure where data are stored or data containing a hierarchical relationship b/w elements.

A **binary** tree + is defined as a finite set of elements called nodes.



## Traversing Binary Trees

There are three ways of traversing a binary tree T with root R.

### Preorder

- 1). Process the root R
- 2). Traverse the left subtree of R in Preorder
- 3). Traverse the Right subtree in preorder

### Inorder

- 1). Traverse left subtree
- 2). Process the root R
- 3). Traverse Right subtree

### Postorder

- 1). Traverse left subtree.
- 2). Traverse Right subtree.
- 3). Process the root R.



# GRAPH

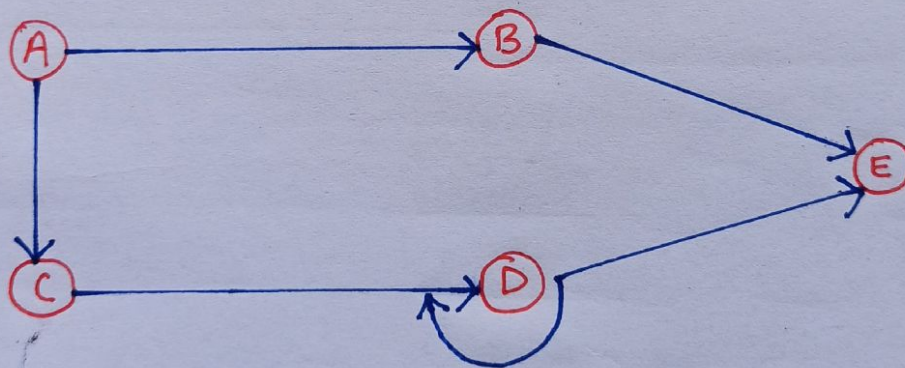
Graph is a collection of two set V and E where,

V  $\longrightarrow$  Vertices / Nodes  
E  $\longrightarrow$  Edges

Graph is a mathematical structures that represent pair-wise relationship between objects where nodes are connected with edges.

Vertex  $\longrightarrow$  Vertex is nothing but the data element which is also known as **Nodes**

Edge  $\longrightarrow$  Edge is a connection link between two vertices



## Representation of the graph

- (A) Adjacency Matrix
- (B) Adjacency List