

# **DATA STRUCTURE**

**Submitted to:- Naveen Dogra Sir**

**Presented by:- Vivek kumar  
Roll no:- SG23340  
Branch:-CSE(2nd Sem)**

# INDEX

- About Data structure
- Need of data structure
- Type if data structure
- Array
- Linked list
- Stack



# ABOUT DATA STRUCTURE

Data structures are a specific way of organizing data in a specialized format on a computer so that the information can be organized, processed, stored, and retrieved quickly and effectively. They are a means of handling information, rendering the data for easy use.

It can hold multiple types of data within a single object. In data structure objects, time complexity plays an important role.

Data structure examples are stack, queue, tree, etc.

Program=Algorithem + Data Structure

# NEED OF DATA STRUCTURE

The structure of the data and the synthesis of the algorithm are relative to each other. Data presentation must be easy to understand so the developer, as well as the user, can make an efficient implementation of the operation. Data structures provide an easy way of organizing, retrieving, managing, and storing data. Here is a list of the needs for data.

Data structure modification is easy.  
It requires less time.  
Save storage memory space.  
Data representation is easy.  
Easy access to the large databases

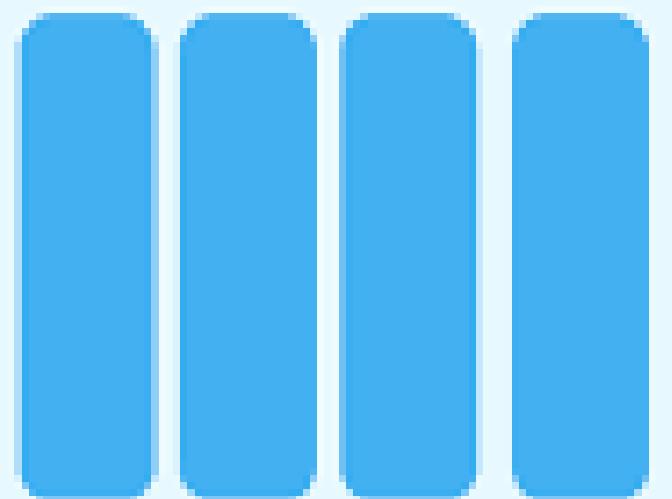
# USES OF DATA STRUCTURE

Different Uses of an Data structure are as follows:

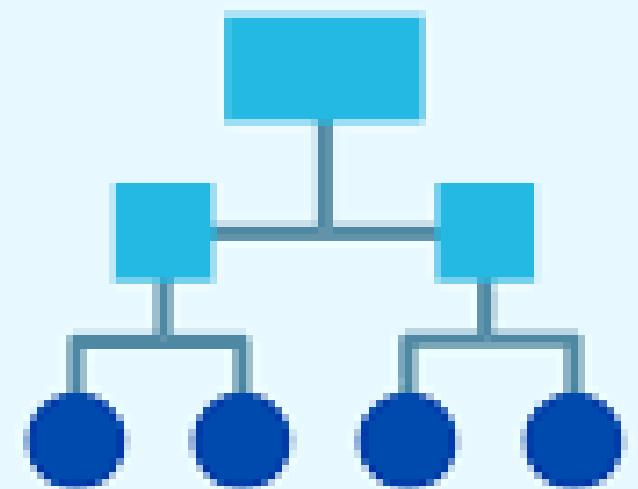
- **Sorting Algorithms:** Lists are essential to constructing efficient sorting algorithms such as quick sort and merge sort.
- **Data Analytics:** Lists are often used to represent datasets in data analytics and machine learning.
- Data structures bring together the data elements in a logical way and facilitate the effective use, persistence and sharing of data.
- They provide a formal model that describes the way the data elements are organized.

# TYPE OF DATA STRUCTURE

## Types of Data Structure

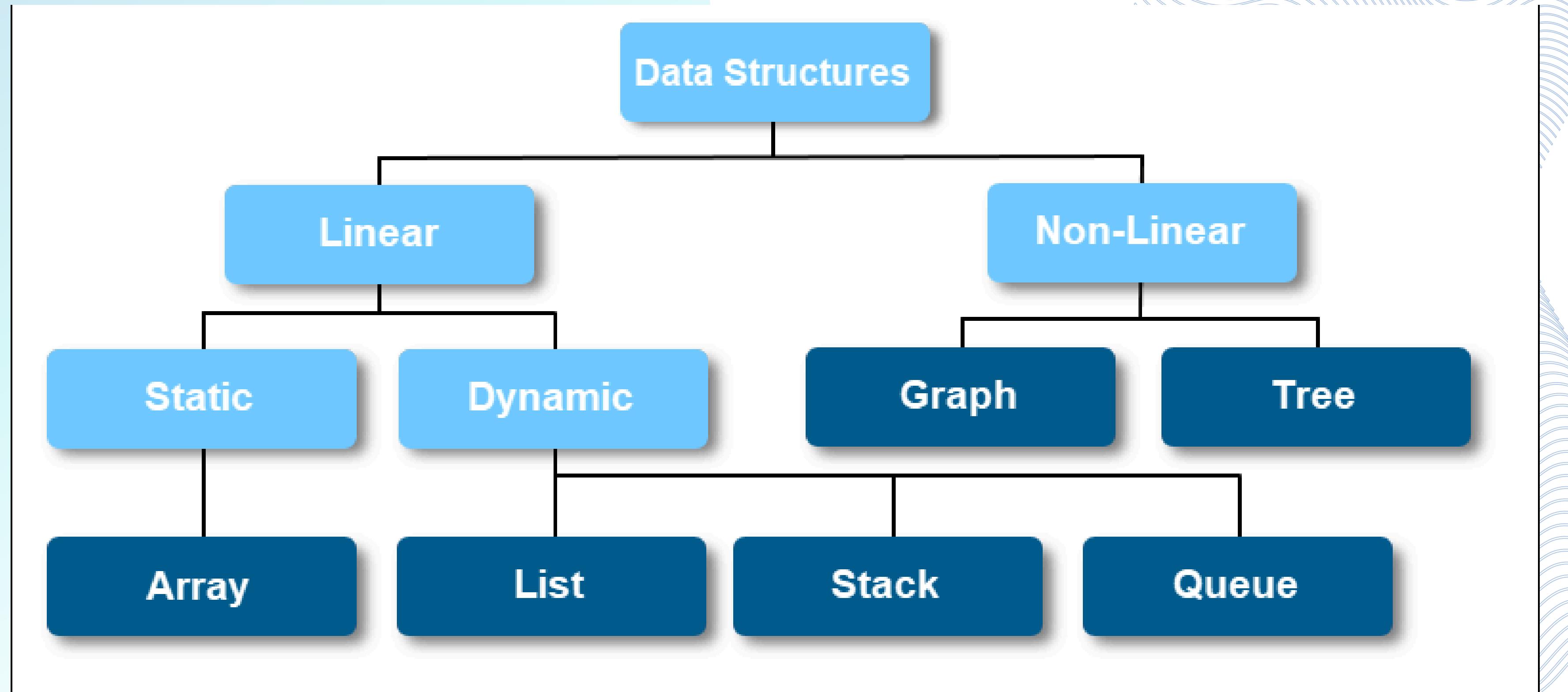


Linear Data Structure



Non Linear Data Structure

# TYPE OF DATA STRUCTURE



# Linear data structure

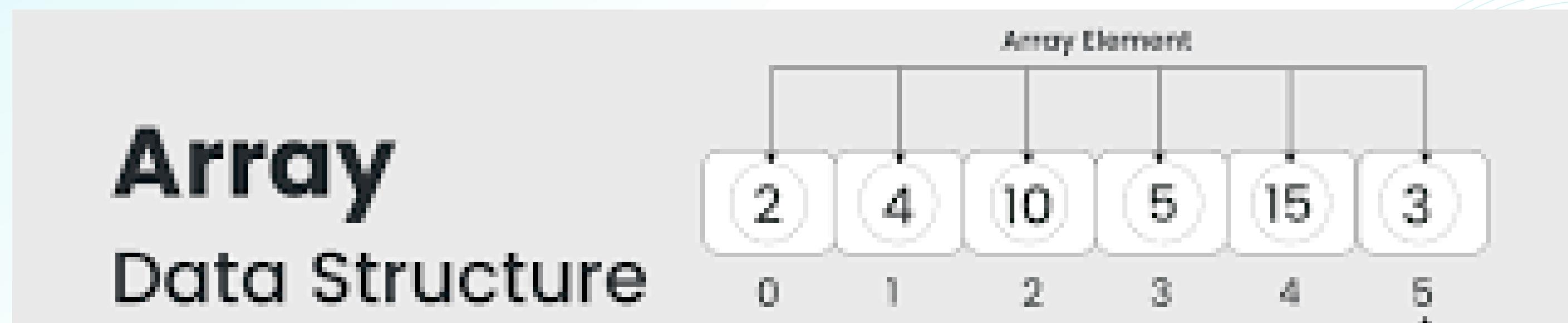
- **Linear data structure:** Data structure in which data elements are arranged sequentially or linearly, where each element is attached to its previous and next adjacent elements, is called a linear data structure.
- Examples of linear data structures are array, stack, queue, linked list, etc.
  - **Static data structure:** Static data structure has a fixed memory size. It is easier to access the elements in a static data structure.
  - An example of this data structure is an array.
  - **Dynamic data structure:** In the dynamic data structure, the size is not fixed. It can be randomly updated during the runtime which may be considered efficient concerning the memory (space) complexity of the code.
  - Examples of this data structure are queue, stack, etc.

# Non linear data structure

- Non-linear data structure: Data structures where data elements are not placed sequentially or linearly are called non-linear data structures. In a non-linear data structure,
- we can't traverse all the elements in a single run only
- . Examples of non-linear data structures are trees and graphs

# ARRAY

- An array is a linear data structure and it is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together in one place. It allows the processing of a large amount of data in a relatively short period. The first element of the array is indexed by a subscript of 0. There are different operations possible in an array, like Searching, Sorting, Inserting, Traversing, Reversing, and Deleting.



# **Characteristics of an Array:**

An array has various characteristics which are as follows:

- Arrays use an index-based data structure which helps to identify each of the elements in an array easily using the index.
- If a user wants to store multiple values of the same data type, then the array can be utilized efficiently.
- An array can also handle complex data structures by storing data in a two-dimensional array.

# **Operations performed on array:**

- Initialization, Searching for elements, Sorting elements, Inserting elements, etc

## **Applications of Array:**

Different applications of an array are as follows:

- An array is used in solving matrix problems.
- Database records are also implemented by an array.
- It helps in implementing a sorting algorithm.

## **Syntax of Array:**

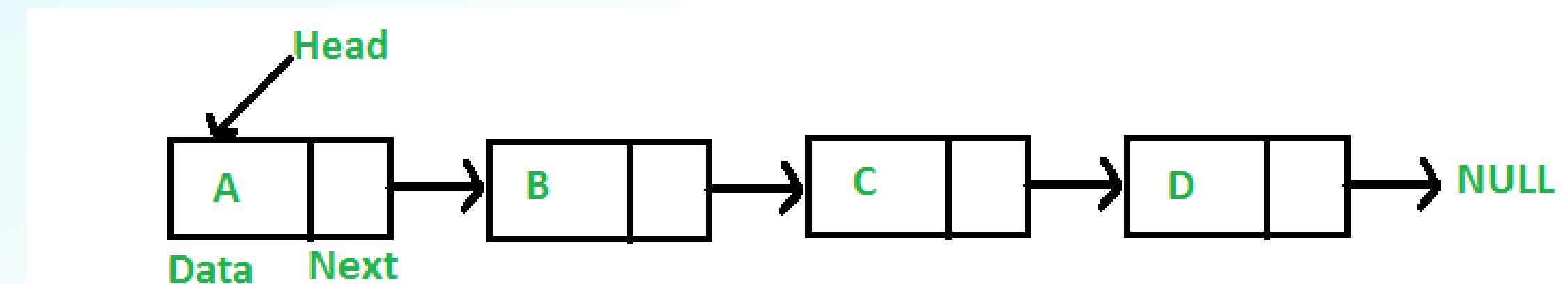
- `datatype Array_Name[size] = { value1, value2, value3, .... valueN };`

# LINKED LIST

A linked list is a linear data structure in which elements are not stored at contiguous memory locations.

Types of linked lists:

- Singly-linked list
- Doubly linked list
- Circular linked list
- Doubly circular linked list



# **Characteristics of a Linked list:**

A linked list has various characteristics which are as follows:

- A linked list uses extra memory to store links.
- During the initialization of the linked list, there is no need to know the size of the elements.
- Linked lists are used to implement stacks, queues, graphs, etc.
- The first node of the linked list is called the Head.
- The next pointer of the last node always points to NULL.

# **Operations performed on Linked list**

- Initialization, Inserting elements, Deleting elements, Searching for elements:

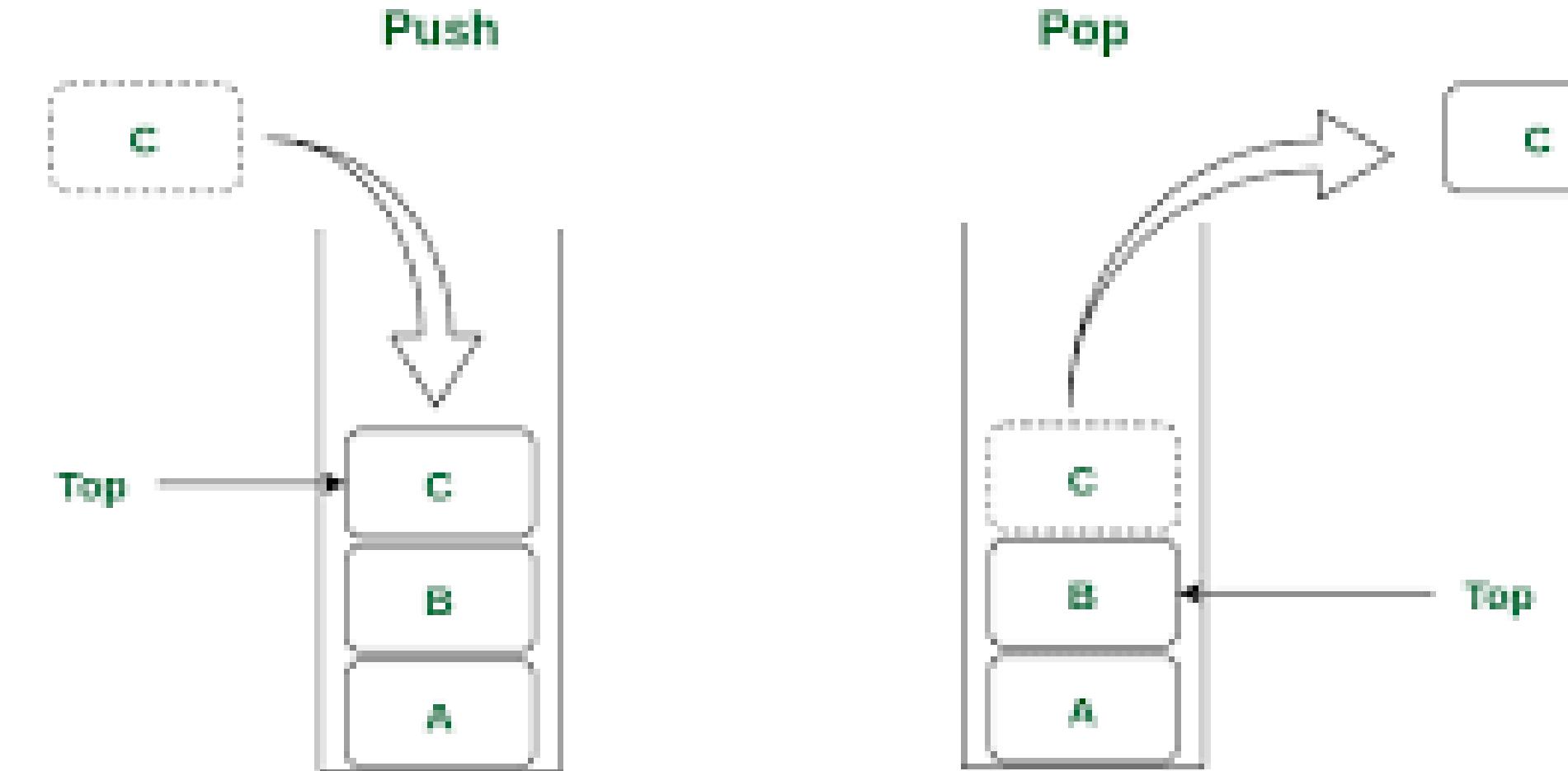
# **Applications of a Linked list:**

Different applications of linked lists are as follows:

- Linked lists are used to implement stacks, queues, graphs, etc.
- Linked lists are used to perform arithmetic operations on long integers.
- It is used for the representation of sparse matrices.
- It is used in the linked allocation of files.
- It helps in memory management.
- It is used in the representation of Polynomial Manipulation where each polynomial term represents a node in the linked list.
- Linked lists are used to display image containers. Users can visit past, current, and next images.

# STACK

STACK IS A LINEAR DATA STRUCTURE THAT FOLLOWS A PARTICULAR ORDER IN WHICH THE OPERATIONS ARE PERFORMED. THE ORDER IS LIFO(LAST IN FIRST OUT). ENTERING AND RETRIEVING DATA IS POSSIBLE FROM ONLY ONE END. THE ENTERING AND RETRIEVING OF DATA IS ALSO CALLED PUSH AND POP OPERATION IN A STACK.



Stack Data Structure

## **Characteristics of a stack:**

**Stack has various different characteristics which are as follows:**

- Stack is used in many different algorithms like Tower of Hanoi, tree traversal, recursion, etc.
- Stack is implemented through an array or linked list.
- It follows the Last In First Out operation i.e., an element that is inserted first will pop in last and vice versa.
- In stack, if the allocated space for the stack is full, and still anyone attempts to add more elements, it will lead to stack overflow.

## **Operations performed on stack**

- Initialization, Inserting elements, Deleting elements, Searching for elements,
- push ,pop, peek

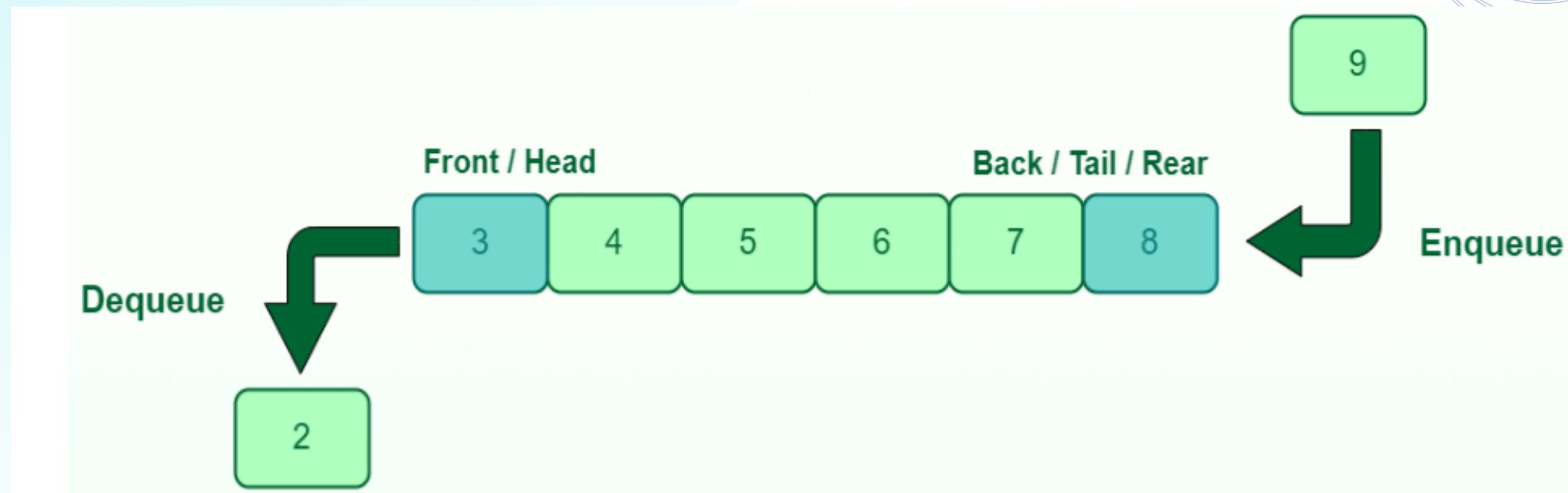
# **Applications of a stack:**

Different applications of Stack are as follows:

- The stack data structure is used in the evaluation and conversion of arithmetic expressions.
- It is used for parenthesis checking.
- While reversing a string, the stack is used as well.
- Stack is used in memory management.
- It is also used for processing function calls.
- The stack is used to convert expressions from infix to postfix.
- The stack is used to perform undo as well as redo operations in word processors.
- The stack is used in virtual machines like JVM.

# QUEUE

QUEUE IS A LINEAR DATA STRUCTURE THAT FOLLOWS A PARTICULAR ORDER IN WHICH THE OPERATIONS ARE PERFORMED. THE ORDER IS FIRST IN FIRST OUT(FIFO) I.E. THE DATA ITEM STORED FIRST WILL BE ACCESSED FIRST. IN THIS, ENTERING AND RETRIEVING DATA IS NOT DONE FROM ONLY ONE END. AN EXAMPLE OF A QUEUE IS ANY QUEUE OF CONSUMERS FOR A RESOURCE WHERE THE CONSUMER THAT CAME FIRST IS SERVED FIRST.



# Characteristics of a queue:

The queue has various different characteristics which are as follows:

- The queue is a FIFO (First In First Out) structure.
- To remove the last element of the Queue, all the elements inserted before the new element in the queue must be removed.
- A queue is an ordered list of elements of similar data types.

# Operations performed on queue

- Enqueue
- Dequeue
- Peek
- IsEmpty
- Size

# Applications of a queue:

Different applications of Queue are as follows:

- Queue is used for handling website traffic.
- It helps to maintain the playlist in media players.
- Queue is used in operating systems for handling interrupts.
- It helps in serving requests on a single shared resource, like a printer, CPU task scheduling, etc.
- It is used in the asynchronous transfer of data e.g. pipes, file IO, and sockets.
- Queues are used for job scheduling in the operating system.
- In social media to upload multiple photos or videos queue is used.
- To send an e-mail queue data structure is used.

# **THANK YOU**

Source:- internet and shared file