**Table of Contents**

* [What is Data Structure?](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#datastructure)
* [How Data Structure varies from Data Type?](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#dsvsdatatype)
* [Classification of Data Structure](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#classification)
  + [Linear Data Structure](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#linear)
    - [1) Static data structure](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#static)
    - [2) Dynamic data structure](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#dynamic)
  + [Non-linear Data Structure](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#nonlinear)
* [Arrays](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#array)
  + [Characteristics of an Array](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#arraychar)
  + [Applications of Array](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#arrayapp)
  + [Real-Life Applications of Array](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#arrayapprl)
* [Linked List](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#linkedlist)
  + [Characteristics of a Linked List](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#llchar)
  + [Applications of Linked List](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#llapp)
  + [Real-Life Applications of Linked List](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#llapprl)
* [Stack](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#stack)
  + [Characteristics of a Stack](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#stackchar)
  + [Applications of Stack](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#stackapp)
  + [Real-Life Applications of Stack](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#stackapprl)
* [Queue](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#queue)
  + [Characteristics of a Queue](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#queuechar)
  + [Applications of Queue](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#queueapp)
  + [Real-Life Applications of Queue](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#queueapprl)
* [Tree](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#tree)
  + [Characteristics of a Tree](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#treechar)
  + [Applications of Tree](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#treeapp)
  + [Real-Life Applications of Tree](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#treeapprl)
* [Graph](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#graph)
  + [Characteristics of a Graph](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#graphchar)
  + [Applications of Graph](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#graphapp)
  + [Real-Life Applications of Graph](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#graphapprl)
* [Conclusion](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/?ref=lbp#conclusion)

# What is Data Structure: Types, Classifications and Applications?

## **What is Data Structure:**

*A data structure is a storage that is used to store and organize data. It is a way of arranging data on a computer so that it can be accessed and updated efficiently.*

A data structure is not only used for organizing the data. It is also used for processing, retrieving, and storing data. There are different basic and advanced types of data structures that are used in almost every program or software system that has been developed. So we must have good knowledge about data structures.

**Data structures** are an integral part of computers used for the arrangement of data in memory. They are essential and responsible for organizing, processing, accessing, and storing data efficiently. But this is not all. Various types of data structures have their own characteristics, features, applications, advantages, and disadvantages. So how do you identify a data structure that is suitable for a particular task? What is meant by the term ‘Data Structure’? How many types of data structures are there and what are they used for?

We have got you covered. We have made a complete list of everything about what data structure is, what are the types of data structures, the classification of data structures, the applications of each data structure, and so on. In this article, we will discuss every aspect of each data structure to help you choose the best one in just minutes.

# **How Data Structure varies from Data Type:**

We already have learned about data structure. Many times, what happens is that people get confused between data type and data structure. So let’s see a few differences between data type and data structure to make it clear.

| Data Type | Data Structure |
| --- | --- |
| The data type is the form of a variable to which a value can be assigned. It defines that the particular variable will assign the values of the given data type only. | Data structure is a collection of different kinds of data. That entire data can be represented using an object and can be used throughout the program. |
| It can hold value but not data. Therefore, it is dataless. | It can hold multiple types of data within a single object. |
| The implementation of a data type is known as abstract implementation. | Data structure implementation is known as concrete implementation. |
| There is no time complexity in the case of data types. | In data structure objects, time complexity plays an important role. |
| In the case of data types, the value of data is not stored because it only represents the type of data that can be stored. | While in the case of data structures, the data and its value acquire the space in the computer’s main memory. Also, a data structure can hold different kinds and types of data within one single object. |
| Data type examples are int, float, double, etc. | Data structure examples are stack, queue, tree, etc. |

# **Classification of Data Structure:**

Data structure has many different uses in our daily life. There are many different data structures that are used to solve different mathematical and logical problems. By using data structure, one can organize and process a very large amount of data in a relatively short period. Let’s look at different data structures that are used in different situations.



* **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 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:**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.*

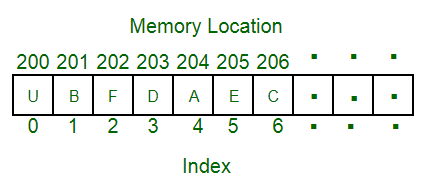
# Need Of Data structure:

# 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.

1. Data structure modification is easy.
2. It requires less time.
3. Save storage memory space.
4. Data representation is easy.
5. Easy access to the large database.

# ****Arrays:****

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.
* An array is also used to implement other data structures like Stacks, Queues, Heaps, Hash tables, etc.
* The search process in an array can be done very easily.

# ****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.
* It is also used to implement other data structures like Stacks, Queues, Heaps, Hash tables, etc.
* An array can be used for CPU scheduling.
* Can be applied as a lookup table in computers.
* Arrays can be used in speech processing where every speech signal is an array.
* The screen of the computer is also displayed by an array . Here we use a multidimensional array.
* The array is used in many management systems like a library, students, parliament, etc.
* The array is used in the online ticket booking system. Contacts on a cell phone are displayed by this array.
* In games like online chess, where the player can store his past moves as well as current moves. It indicates a hint of position.
* To save images in a specific dimension in the android Like 360\*1200

# ****Real-Life Applications of Array:****

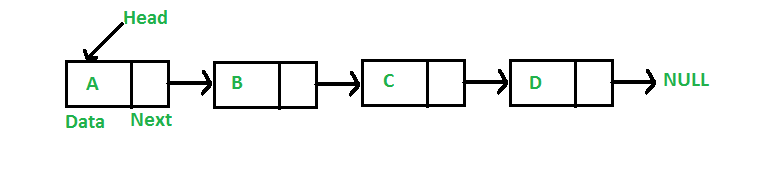
* An array is frequently used to store data for mathematical computations.
* It is used in image processing.
* It is also used in record management.
* Book pages are also real-life examples of an array.
* It is used in ordering boxes as well.

# ****Linked list:****

A linked list is a linear data structure in which elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers as shown in the below image:

Types of linked list:

* 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 initialization of 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.
* In linked list, insertion and deletion is possible easily.
* Each node of the linked list consists of a pointer/link which is the address of the next node.
* Linked list can shrink or grow at any point in time easily.

# ****Applications of the Linked list:****

### 

Different applications of linked list are as follows:

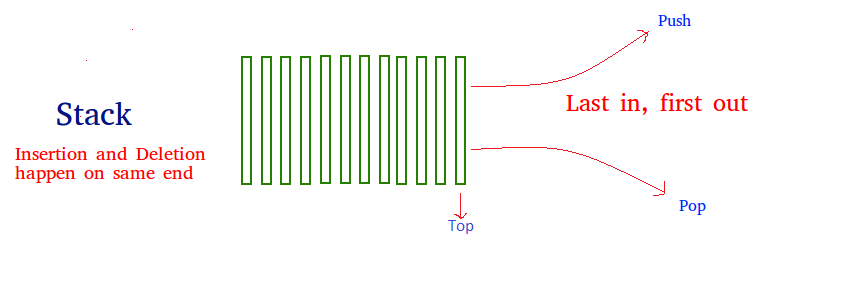
* Linked lists are used to implement stacks, queues, graphs, etc.
* Linked lists are used to perform arithmetic operations on long integers.
* It is used for the representation of sparse matrices.
* It is used in linked allocation of files.
* It helps in memory management.
* It is used in the representation 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.
* They are used to store the history of the visited page.
* They are used to perform undo operations.
* Linked are used in software development where they indicate the correct syntax of a tag.
* Linked lists are used to display social media feeds.

# ****Real-Life Applications of a Linked list:****

* A linked list is used in Round-Robin scheduling to keep track of the turn in multi-player games.
* It is used in image viewer. The previous and next images are linked, hence can be accessed by the previous and next buttons.
* In a music playlist, songs are linked to the previous and next songs.

# ****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)](https://www.geeksforgeeks.org/lifo-last-in-first-out-approach-in-programming/). 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. There are different operations possible in a stack like reversing a stack using recursion, Sorting, Deleting the middle element of a stack, etc.



# ****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 array or linked list.
* It follows Last In First Out operation i.e., element which is inserted first will pop in last and vice versa.
* The insertion and deletion happens at one end i.e. from the top of the stack.
* In stack, if allocated space for stack is full, and still anyone attempts to add more elements, it will lead to stack overflow.

# ****Applications of Stack:****

Different applications of Stack are as follows:

* Stack data structure is used in evaluation and conversion of arithmetic expressions.
* Stack is used in Recursion.
* It is used for parenthesis checking.
* While reversing a string, stack is used as well.
* Stack is used in memory management.
* It is also used for processing of 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.
* The stack is used in the media players. Useful to play the next and previous song.
* The stack is used in recursion operation.

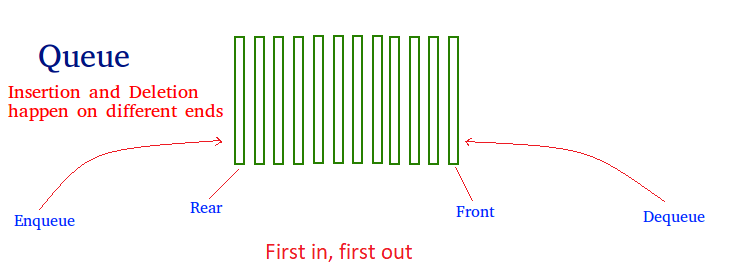
### **Real Life Applications of Stack:**

* Real life example of a stack is the layer of eating plates arranged one above the other. When you remove a plate from the pile, you can take the plate on the top of the pile. But this is exactly the plate that was added most recently to the pile. If you want the plate at the bottom of the pile, you must remove all the plates on top of it to reach it.
* Browsers use stack data structure to keep track of previously visited sites.
* Call log in mobile also uses stack data structure.

Want to get started with Stack? You can try out our curated articles and lists for the best practice

# ****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)](https://www.geeksforgeeks.org/fifo-first-in-first-out-approach-in-programming/) 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. Different operations are performed on Queue like Reversing a Queue (with or without using recursion), Reversing the first K elements of a Queue, etc. Few basic operations performed In Queue are enqueue, dequeue, front, rear, etc.



# ****Characteristics of a Queue:****

Queue has various different characteristics which are as follows:

* Queue is a FIFO (First In First Out) structure.
* To remove the last element of 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.

# ****Applications of 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 asynchronous transfer of data for e.g. pipes, file IO, sockets.
* Queues are used for job scheduling in operating system.
* In social media to upload multiple phots or videos queue is used.
* To send an e-mail queue data structure is used.
* To handle website traffic at a time queue are used.
* In window operating system, to switch multiple application.

# ****Real-Life Applications of Queue:****

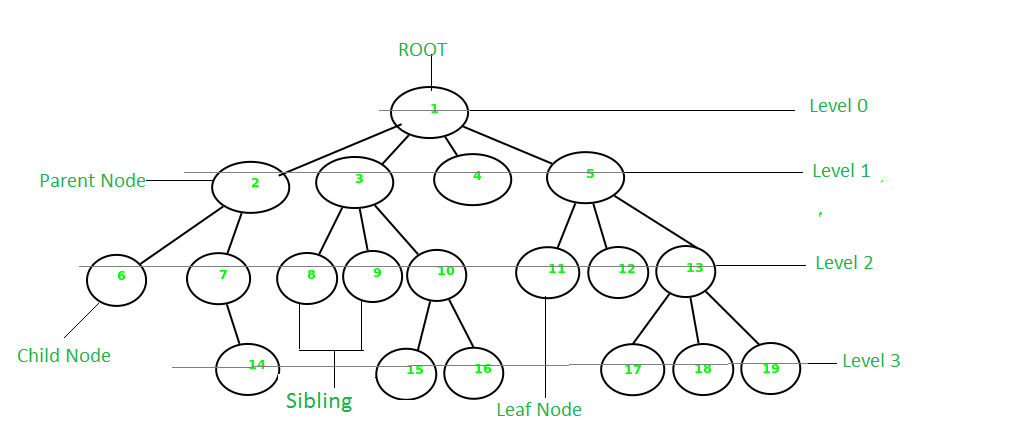
* A real world example of queue is a single lane one way road, where the vehicle that enters first will exit first.
* A more real-world example can be seen in the queue at the ticket windows.
* Cashier line in a store is also an example of queue.
* People on an escalator

# Tree:

A tree is a non-linear and hierarchal data structure where the elements are arranged in a tree-like structure. In a tree, the topmost node is called the root node. Each node contains some data, and data can be of any type. It consists of a central node, structural nodes, and sub-nodes which are connected via edges. Different tree data structures allow quicker and easier access to the data as it is a non-linear data structure. A tree has various terminologies like Node, Root, Edge, Height of a tree, Degree of a tree, etc.

There are different types of Tree like

* [Binary Tree](https://www.geeksforgeeks.org/binary-tree-data-structure/),
* [Binary Search Tree](http://www.geeksforgeeks.org/binary-search-tree-set-1-search-and-insertion/),
* [AVL Tree](https://www.geeksforgeeks.org/avl-tree-set-1-insertion/),
* [B-Tree,](https://www.geeksforgeeks.org/introduction-of-b-tree-2/) etc.



# ****Characteristics of a Tree:****

Tree has various different characteristics which are as follows:

* A tree is also known as a Recursive data structure.
* In a tree, Height of the root can be defined as the longest path from the root node to the leaf node.
* In a tree, one can also calculate the depth from the top to any node. The root node has depth of 0.

# ****Applications of Tree:****

Different applications of Tree are as follows:

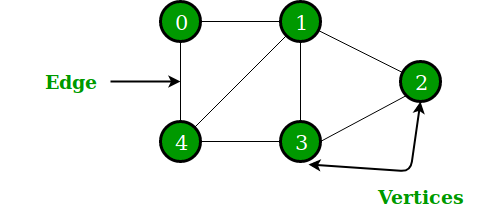
* Heap is a tree data structure that is implemented using arrays, and used to implement priority queues.
* B-Tree and B+ Tree are used to implement indexing in databases.
* Syntax Tree helps in scanning, parsing, generation of code and evaluation of arithmetic expressions in Compiler design.
* K-D Tree is a space partitioning tree used to organize points in K-dimensional space.
* Spanning trees are used in routers in computer networks.

# ****Real Life Applications of Tree:****

* In real life, tree data structure helps in Game Development.
* It also helps in indexing in databases.
* Decision Tree is an efficient machine learning tool, commonly used in decision analysis. It has a flowchart-like structure that helps to understand data.
* Domain Name Server also uses tree data structure.
* The most common use case of a tree is any social networking site.

# Graph:

A graph is a non-linear data structure that consists of vertices (or nodes) and edges. It consists of a finite set of vertices and set of edges that connect a pair of nodes. Graph is used to solve the most challenging and complex programming problems. It has different terminologies which are Path, Degree, Adjacent vertices, Connected components, etc.



# ****Characteristics of Graph:****

Graph has various different characteristics which are as follows:

* The maximum distance from a vertex to all the other vertices is considered as the Eccentricity of that vertex.
* The vertex having minimum Eccentricity is considered the central point of the graph.
* The minimum value of Eccentricity from all vertices is considered as the radius of a connected graph.

# ****Applications of Graph:****

Different applications of Graph are as follows:

* Graph is used to represent the flow of computation.
* It is used in modeling graphs.
* The operating system uses Resource Allocation Graph.
* Also used in the World Wide Web where the web pages represent the nodes.

# ****Real-Life Applications of Graph:****

* One of the most common real-world examples of a graph is Google Maps where cities are located as vertices and paths connecting those vertices are located as edges of the graph.
* A social network is also one real-world example of a graph where every person on the network is a node, and all of their friendships on the network are the edges of the graph.
* A graph is also used to study molecules in physics and chemistry.