

## Data Structures Tutorial

```

      11001
    10001 11100110
  0010 110001 11000110
00101001 01011010 1100
010101 1100000100 100
00011111 10 1001110
  00101 11010 10
    10010 101
      00100
        01001
          00110
            0000110
  
```

## Data Structure

Data Structures (DS) tutorial provides basic and advanced concepts of Data Structure. Our Data Structure tutorial is designed for beginners and professionals.

Data Structure is a way to store and organize data so that it can be used efficiently.

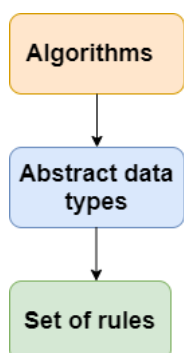
Our Data Structure tutorial includes all topics of Data Structure such as Array, Pointer, Structure, Linked List, Stack, Queue, Graph, Searching, Sorting, Programs, etc.

### What is Data Structure?

The data structure name indicates itself that organizing the data in memory. There are many ways of organizing the data in the memory as we have already seen one of the data structures, i.e., array in C language. Array is a collection of memory elements in which data is stored sequentially, i.e., one after another. In other words, we can say that array stores the elements in a continuous manner. This organization of data is done with the help of an array of data structures. There are also other ways to organize the data in memory. Let's see the different types of data structures.

The data structure is not any programming language like C, C++, java, etc. It is a set of algorithms that we can use in any programming language to structure the data in the memory.

To structure the data in memory, 'n' number of algorithms were proposed, and all these algorithms are known as Abstract data types. These abstract data types are the set of rules.



### Types of Data Structures

There are two types of data structures:

- Primitive data structure
- Non-primitive data structure

### Primitive Data structure

The primitive data structures are primitive data types. The int, char, float, double, and pointer are the primitive data structures that can hold a single value.

### Non-Primitive Data structure

The non-primitive data structure is divided into two types:

- Linear data structure
- Non-linear data structure

### Linear Data Structure

The arrangement of data in a sequential manner is known as a linear data structure. The data structures used for this purpose are Arrays, Linked list, Stacks, and Queues. In these data structures, one element is connected to only one another element in a linear form.

**When one element is connected to the 'n' number of elements known as a non-linear data structure. The best example is trees and graphs. In this case, the elements are arranged in a random manner.**

We will discuss the above data structures in brief in the coming topics. Now, we will see the common operations that we can perform on these data structures.

**Data structures can also be classified as:**

- **Static data structure:** It is a type of data structure where the size is allocated at the compile time. Therefore, the maximum size is fixed.
- **Dynamic data structure:** It is a type of data structure where the size is allocated at the run time. Therefore, the maximum size is flexible.

## Major Operations

The major or the common operations that can be performed on the data structures are:

- **Searching:** We can search for any element in a data structure.
- **Sorting:** We can sort the elements of a data structure either in an ascending or descending order.
- **Insertion:** We can also insert the new element in a data structure.
- **Update:** We can also update the element, i.e., we can replace the element with another element.
- **Deletion:** We can also perform the delete operation to remove the element from the data structure.

## Which Data Structure?

A data structure is a way of organizing the data so that it can be used efficiently. Here, we have used the word efficiently, which in terms of both the space and time. For example, a stack is an ADT (Abstract data type) which uses either arrays or linked list data structure for the implementation. Therefore, we conclude that we require some data structure to implement a particular ADT.

An ADT tells **what** is to be done and data structure tells **how** it is to be done. In other words, we can say that ADT gives us the blueprint while data structure provides the implementation part. Now the question arises: how can one get to know which data structure to be used for a particular ADT?.

As the different data structures can be implemented in a particular ADT, but the different implementations are compared for time and space. For example, the Stack ADT can be implemented by both Arrays and linked list. Suppose the array is providing time efficiency while the linked list is providing space efficiency, so the one which is the best suited for the current user's requirements will be selected.

## Advantages of Data structures

**The following are the advantages of a data structure:**

- **Efficiency:** If the choice of a data structure for implementing a particular ADT is proper, it makes the program very efficient in terms of time and space.
- **Reusability:** The data structures provide reusability means that multiple client programs can use the data structure.
- **Abstraction:** The data structure specified by an ADT also provides the level of abstraction. The client cannot see the internal working of the data structure, so it does not have to worry about the implementation part. The client can only see the interface.

## Data Structures Index

**DS Basics**

- DS Tutorial
- DS Introduction
- DS Algorithm
- DS Asymptotic Analysis
- DS Pointer
- DS Structure

**DS Array**

- Array
- 2D Array

**DS Linked List**

- Linked List
  - Insertion at beginning
  - Insertion at end
  - Insertion after specified node
  - Deletion at beginning
  - Deletion at end
  - Deletion after specified node
  - Traversing
  - Searching
- Doubly Linked List
  - Insertion at beginning
  - Insertion at end
  - Insertion after specified node
  - Deletion at beginning
  - Deletion at end
  - Deletion of node having given data
  - Traversing
  - Searching
- Circular Linked List
  - Insertion at beginning
  - Insertion at end
  - Deletion at beginning
  - Deletion at the end
  - Traversing
  - Searching
- Circular Doubly List
  - Insertion at beginning
  - Insertion at end
  - Deletion at beginning
  - Deletion at the end

**DS Stack**

- DS Stack
- Array Implementation
- Linked List Implementation

**DS Queue**

- DS Queue
- Array Implementation
- Linked List Implementation
- Circular Queue

**DS Tree**

- Tree
- Binary Tree
  - Pre-order Traversal
  - In-order Traversal
  - Post-order Traversal
- Binary Search Tree

**Singly Linked List Programs**

- Program to create and display a singly linked list
- Program to create a singly linked list of n nodes and count the number of nodes
- Program to create a singly linked list of n nodes and display it in reverse order
- Program to delete a new node from the beginning of the singly linked list
- Program to delete a new node from the middle of the singly linked list
- Program to delete a node from the end of the singly linked list
- Program to determine whether a singly linked list is the palindrome
- Program to find the maximum and minimum value node from a singly linked list
- Program to insert a new node at the middle of the singly linked list
- Program to insert a new node at the beginning of the singly linked list
- Program to insert a new node at the end of the singly linked list
- Program to remove duplicate elements from a singly linked list
- Program to search an element in a singly linked list
- Program to sort the elements of the singly linked list
- Program to swap nodes in a singly linked list without swapping data
- Program to swap the last element of the singly linked list from the first one

**Doubly Linked List Programs**

- Program to Convert a Given Binary Tree to Doubly Linked List
- Program to Create a Doubly Linked List From a Ternary Tree
- Program to Create a Doubly Linked List of N Nodes and Count the Number of Nodes
- Program to Create a Doubly Linked List of N Nodes and Display it in Reverse Order
- Program to Create and Display a Doubly Linked List
- Program to Delete a New Node From the Beginning of the Doubly Linked List
- Program to Delete a New Node From the End of the Doubly Linked List
- Program to Delete a New Node From the Middle of the Doubly Linked List
- Program to Find the Maximum and Minimum Value Node From a Doubly Linked List
- Program to Insert a New Node at the Beginning of the Doubly Linked List
- Program to Insert a New Node at the End of Doubly Linked List
- Program to Insert a New Node at the Middle of Doubly Linked List
- Program to Remove Duplicate Elements From a Doubly Linked List
- Program to Rotate Doubly Linked List by N Nodes
- Program to Search an Element in a Doubly Linked List
- Program to Sort the Elements of the Doubly Linked List

**Circular Linked List Programs**

- Program to Create a Circular Linked List of N Nodes and Count the Number of Nodes
- Program to Create a Circular Linked List of N Nodes and Display it in Reverse Order

- Searching in BST
- Insertion in BST
- Deletion in BST
- AVL Tree
  - Insertion in AVL Tree
    - LL Rotation
    - LR Rotation
    - RL Rotation
    - RR Rotation
  - Deletion in AVL Tree
- B Tree
- B+ Tree
- Red Black Tree

**DS Graph**

- DS Graph
- Graph Implementation
- BFS Algorithm
- DFS Algorithm
- Spanning Tree
  - Prim's Algorithm
  - Kruskal's Algorithm

**DS Searching**

- Linear Search
- Binary Search

**DS Sorting**

- Bubble Sort
- Bucket Sort
- Comb Sort
- Counting Sort
- Heap Sort
- Insertion Sort
- Merge Sort
- Quick Sort
- Radix Sort
- Selection Sort
- Shell Sort
- Bitonic Sort
- Cocktail Sort
- Cycle Sort
- Tim Sort

**Interview Questions**

- DS Interview Questions

- Program to Create and Display a Circular Linked List
- Program to Delete a New Node From the Beginning of the Circular Linked List
- Program to Delete a New Node From the End of the Circular Linked List
- Program to Delete a New Node From the Middle of the Circular Linked List
- Program to Find the Maximum and Minimum Value Node From a Circular Linked List
- Program to Insert a New Node at the Beginning of the Circular Linked List
- Program to Insert a New Node at the End of the Circular Linked List
- Program to Insert a New Node at the Middle of the Circular Linked List
- Program to Remove Duplicate Elements From a Circular Linked List
- Program to Search an Element in a Circular Linked List
- Program to Sort the Elements of the Circular Linked List

**Tree Programs**

- Program to Calculate the Difference Between the Sum of the Odd Level and Even Level Nodes of a Binary Tree
- Program to Construct a Binary Search Tree and Perform Deletion and Inorder Traversal
- Program to Convert Binary Tree to Binary Search Tree
- Program to Determine Whether all Leaves are at Same Level
- Program to Determine Whether two Trees are Identical
- Program to Find Maximum Width of a Binary Tree
- Program to Find the Largest Element in a Binary Tree
- Program to Find the Maximum Depth or Height of a Tree
- Program to Find the Nodes Which are at the Maximum Distance in a Binary Tree
- Program to Find the Smallest Element in a Binary Tree
- Program to Find the Sum of all the Nodes of a Binary Tree
- Program to Find the Total Number of Possible Binary Search Trees with N Keys
- Program to Implement Binary Tree using the Linked List
- Program to Search a Node in a Binary Tree

## Prerequisite

Before learning Data Structure, you must have the basic knowledge of C.

## Audience

Our Data Structure tutorial is designed to help beginners and professionals.

## Problem

We assure that you will not find any problem in this Data Structure tutorial. But if there is any mistake, please post it in the contact form.

 [For Videos Join Our Youtube Channel: Join Now](#)

Help Others, Please Share







### Remote American Software Jobs

U.S Software Developers Jobs. Work Remotely. No Visa needed. Full time jobs. High salary.


Learn Latest Tutorials




Apache Solr Tutorial  
Solr




MongoDB tutorial  
MongoDB




Gimp Tutorial  
Gimp




Verilog Tutorial  
Verilog




Teradata Tutorial  
Teradata




PhoneGap Tutorial  
PhoneGap




Gmail Tutorial  
Gmail




Vue.js Tutorial  
Vue.js



PLC tutorial  
PLC




Adobe Illustrator Tutorial  
Illustrator




Postman Tutorial  
Postman


Preparation




Aptitude  
Aptitude




Logical Reasoning  
Reasoning



Verbal Ability  
Verbal A.




Interview Questions  
Interview




Company Interview Questions  
Company


Trending Technologies




AI



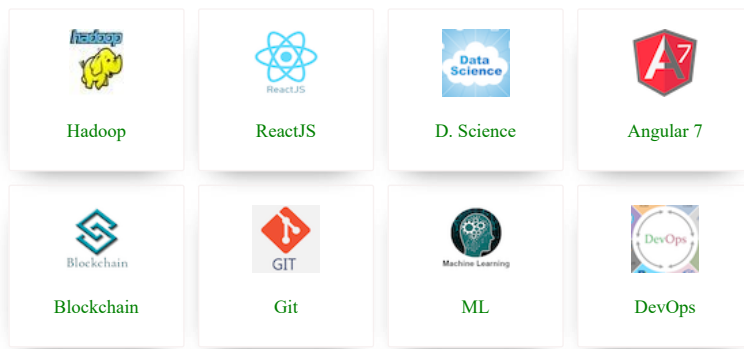
AWS



Selenium



Cloud



## B.Tech / MCA

