

An **array data structure** is a fundamental concept in computer science that stores a collection of elements in a contiguous block of memory. It allows for efficient access to elements using indices and is widely used in programming for organizing and manipulating data.

What is an Array?

An **array** is a collection of items of the same variable type that are stored at contiguous memory locations. It's one of the most popular and simple data structures and is often used to implement other data structures. Each item in an array is indexed starting with **0**. Each element in an array is accessed through its index.

Need of Array Data Structures

Arrays are a fundamental data structure in computer science. They are used in a wide variety of applications, including:

- Storing data for processing
- Implementing data structures such as stacks and queues
- Representing data in tables and matrices
- Creating dynamic data structures such as linked lists and trees

Types of Arrays

There are two main types of arrays:

- **One-dimensional arrays:** These arrays store a single row of elements.
- **Multidimensional arrays:** These arrays store multiple rows of elements.

Array Operations

Common operations performed on arrays include:

- **Traversal:** Visiting each element of an array in a specific order (e.g., sequential, reverse).
- **Insertion:** Adding a new element to an array at a specific index.
- **Deletion:** Removing an element from an array at a specific index.
- **Searching:** Finding the index of an element in an array.

Applications of Array

Arrays are used in a wide variety of applications, including:

- Storing data for processing
- Implementing data structures such as stacks and queues
- Representing data in tables and matrices
- Creating dynamic data structures such as linked lists and trees

Advantages of Array

- Arrays allow random access to elements. This makes accessing elements by position faster.
- Arrays have better cache locality which makes a pretty big difference in performance.
- Arrays represent multiple data items of the same type using a single name.
- Arrays are used to implement the other data structures like linked lists, stacks, queues, trees, graphs, etc.

Complexity Analysis of Operations on Array

Time Complexity:

Operation	Best Case	Average Case	Worst Case
Traversal	$\Omega(N)$	$\theta(N)$	$O(N)$
Insertion	$\Omega(1)$	$\theta(N)$	$O(N)$
Deletion	$\Omega(1)$	$\theta(N)$	$O(N)$
Searching	$\Omega(1)$	$\theta(N)$	$O(N)$

Space Complexity:

Operation	Best Case	Average Case	Worst Case
Traversal	$\Omega(1)$	$\theta(1)$	$O(1)$
Insertion	$\Omega(1)$	$\theta(N)$	$O(N)$
Deletion	$\Omega(1)$	$\theta(N)$	$O(N)$
Searching	$\Omega(1)$	$\theta(1)$	$O(1)$

Disadvantages of Array

- As arrays have a fixed size, once the memory is allocated to them, it cannot be increased or decreased, making it impossible to store extra data if required. An array of fixed size is referred to as a static array.
- Allocating less memory than required to an array leads to loss of data.
- An array is homogeneous in nature so, a single array cannot store values of different data types.
- Arrays store data in contiguous memory locations, which makes deletion and insertion very difficult to implement. This problem is overcome by implementing linked lists, which allow elements to be accessed sequentially.

