### **How Arrays are Represented in Memory**

An array is a contiguous block of memory locations used to store elements of the same data type. Each element in the array is accessed using its index, which is an integer representing the offset from the base address of the array.

### **Advantages of Arrays**

* Simple to implement and understand.
* Efficient for storing and accessing elements.
* Suitable for data structures that require direct access to elements.

### **Time Complexity**

* **Add:** O(1) - Constant time, if there's space available.
* **Search:** O(n) - Linear time, as we need to iterate through the array in the worst case.
* **Traverse:** O(n) - Linear time, as we need to visit each element.
* **Delete:** O(n) - Linear time, due to shifting elements.

### **Limitations of Arrays**

* Fixed size: Cannot dynamically resize.
* Inefficient for insertions and deletions at arbitrary positions due to shifting elements.

### **When to Use Arrays**

* When the number of elements is known beforehand.
* When random access is the primary operation.
* When memory efficiency is critical and the size of the data is relatively small.