**Arrays**:

* **Contiguous Allocation**: Arrays store elements in contiguous memory locations, making them efficient for indexed access.
* **Advantages**: Fast Access - O(1) time complexity for accessing elements via index.

**Time Complexity**:

* **Add**: O(1)
* **Search**: O(n)
* **Traverse**: O(n)
* **Delete**: O(n)

**Limitations of Arrays**:

* **Fixed Size**: Cannot dynamically grow or shrink, which can lead to wasted space or lack of space.
* **Costly Insertions and Deletions**: Inserting or deleting an element can be costly as it may involve shifting other elements.

**When to Use Arrays**:

* Arrays are suitable when the number of elements is known in advance and efficient indexed access is needed.