**Understanding Array Representation**

**Array Representation in Memory**:

* Arrays are a collection of elements stored in contiguous memory locations.
* Each element in the array can be accessed using its index, which allows for constant-time access (O(1)).

**Advantages**:

* + Fast access to elements using indices.
  + Efficient use of memory due to contiguous storage.
  + Simple and easy to implement.

**Analysis**

**Time Complexity**:

**Add**: O(1) - Adding an employee to the end of the array is a constant-time operation.

**Search**: O(n) - In the worst case, we may need to search through all employees.

**Traverse**: O(n) - We need to visit each employee once.

**Delete**: O(n) - In the worst case, we may need to search through all employees to find the one to delete.

**Limitations of Arrays**:

Fixed size: Once an array is created, its size cannot be changed.

Inefficient for dynamic data: Adding or removing elements can be inefficient if the array is full or if elements need to be shifted.

When to use: Arrays are suitable when the number of elements is known in advance and does not change frequently.