**Exercise 4: Employee Management System**

**1. Understand Array Representation**

**1.1 Explain How Arrays are Represented in Memory and Their Advantages**

* **Memory Representation**:
  + Arrays are contiguous blocks of memory where each element is of the same data type.
  + The memory address of any element can be calculated using the base address and the index of the element.
* **Advantages**:
  + **Direct Access**: Arrays allow direct access to elements using the index, making retrieval operations very fast (O(1) time complexity).
  + **Cache-Friendly**: Due to contiguous memory allocation, arrays are cache-friendly and lead to faster data access.
  + **Fixed Size**: The fixed size of arrays ensures that memory allocation is efficient and predictable

**4. Analysis**

**Step 4.1: Analyze the Time Complexity of Each Operation**

* **Add**: O(1) (assuming there is space available in the array)
* **Search**: O(n) (linear search)
* **Traverse**: O(n)
* **Delete**: O(n) (worst-case, linear search to find the element)

**Step 4.2: Discuss the Limitations of Arrays and When to Use Them**

* **Limitations**:
  + **Fixed Size**: Arrays have a fixed size, which can lead to wasted memory if the array is too large or insufficient memory if the array is too small.
  + **Insertion and Deletion**: Adding or deleting elements can be inefficient because elements may need to be shifted.
  + **Static Nature**: Arrays cannot grow or shrink dynamically, making them less flexible than other data structures like ArrayList or LinkedList.
* **When to Use Arrays**:
  + Arrays are ideal for situations where the number of elements is known and does not change frequently.
  + Arrays are suitable for applications that require fast access to elements by index.
  + Use arrays when memory usage and allocation predictability are important.