**Exercise 4: Employee Management System**

**Scenario:**

You are developing an employee management system for a company. Efficiently managing employee records is crucial.

**Understanding Array Representation:**

Arrays are a collection of elements stored in contiguous memory locations. Each element in the array can be accessed directly using its index, which makes arrays efficient for read operations.

**Advantages of Arrays:**

1. **Fast Access**: Direct access to elements using indices provides constant time complexity (O(1)) for read operations.
2. **Predictable Memory Usage**: Fixed size allows for predictable memory allocation.
3. **Ease of Traversal**: Sequential storage simplifies iteration over elements.

However, arrays have limitations such as fixed size (inflexibility in size adjustment) and costly operations for adding or deleting elements (except at the end).

**Analysis:**

1. **Add Operation**:
   * **Time Complexity**: O(1) - Adding an element to the end of the array is a constant time operation.
2. **Search Operation**:
   * **Time Complexity**: O(n) - Linear search is required, so in the worst case, it has to check all elements.
3. **Traverse Operation**:
   * **Time Complexity**: O(n) - Each element is visited once.
4. **Delete Operation**:
   * **Time Complexity**: O(n) - Worst-case scenario involves searching for the element, then shifting elements.

**Limitations of Arrays:**

1. **Fixed Size**: Once an array is created, its size cannot be changed. This can lead to either wasted space or insufficient capacity.
2. **Inefficient Insertions/Deletions**: Adding or removing elements (except at the end) requires shifting elements, which can be costly (O(n)).
3. **Memory Contiguity**: Requires a contiguous block of memory, which can be a limitation for large arrays.

**When to Use Arrays:**

* When the size of the collection is known and fixed.
* When fast read access is required.
* When the overhead of dynamic data structures like linked lists or array lists is not justified.