**Exercise 4: Employee Management System**

**Explain how arrays are represented in memory and their advantages.**

**Array Representation in Memory**

An array is a collection of elements stored in contiguous memory locations. The elements can be accessed randomly using indices. Arrays offer efficient access and modification times due to their contiguous memory layout.

**Advantages of Arrays**:

1. **Random Access**: Accessing any element using its index is O(1).
2. **Memory Efficiency**: Arrays have minimal overhead compared to other data structures like linked lists.
3. **Cache Friendliness**: Due to their contiguous memory allocation, arrays are cache-friendly, leading to performance improvements.

**Analyze the time complexity of each operation (add, search, traverse, delete).**

Time Complexity of Operations

1. Add Employee: O(1) - Adding an employee to the end of the array is constant time if the array is not full.
2. Search Employee: O(n) - In the worst case, we may need to search through all n employees.
3. Traverse Employees: O(n) - We need to visit each employee once.
4. Delete Employee: O(n) - In the worst case, we may need to search through all n employees to find the one to delete.

**Discuss the limitations of arrays and when to use them.**

Limitations of Arrays

1. Fixed Size: Arrays have a fixed size, which must be defined at creation. This can lead to wasted memory if the array is not fully utilized or insufficient capacity if the array is too small.
2. Insertion and Deletion: While adding or deleting at the end of the array is O(1), inserting or deleting in the middle requires shifting elements, which is O(n).
3. Memory Contiguity: Arrays require a contiguous block of memory, which can be a limitation for large datasets.

When to Use Arrays

1. When random access is required: Arrays provide constant-time access to elements using their indices.
2. When memory efficiency is important: Arrays have minimal overhead.
3. For static datasets: When the size of the dataset is known and does not change frequently.