**Exercise 4: Employee Management System**

**1. Understand Array Representation**

**Array Representation in Memory**

* **Contiguous Memory Allocation**: Arrays are stored in contiguous memory locations. Each element of the array is placed sequentially in memory.
* **Indexing**: Elements in an array can be accessed using their index. The index of the first element is 0, the second element is 1, and so on.
* **Advantages**:
  + **Fast Access**: Direct access to elements using their index, making read operations O(1).
  + **Memory Efficiency**: Arrays are memory efficient as they don't require additional memory for pointers or node structures.

**4. Analysis**

**Time Complexity of Operations**

* **Add**: O(1) - Adding an employee to the end of the array is a constant-time operation.
* **Search**: O(n) - In the worst case, the search operation needs to check all elements.
* **Traverse**: O(n) - Traversing the array requires visiting each element once.
* **Delete**: O(n) - In the worst case, finding the element to delete requires O(n) time, and removing it involves swapping the last element to the position of the deleted element.

**Limitations of Arrays**

* **Fixed Size**: Arrays have a fixed size, meaning the maximum number of elements is determined when the array is created. This can lead to wasted memory or insufficient space.
* **Insertion and Deletion Overhead**: Inserting or deleting elements in the middle of an array can be costly, as it may require shifting elements.
* **Sequential Access**: While direct access by index is efficient, searching for an element requires sequential access, leading to O(n) time complexity.

**When to Use Arrays**

* **Static Data**: When the number of elements is known in advance and does not change.
* **Fast Access**: When fast, direct access to elements is required.
* **Memory Efficiency**: When memory overhead needs to be minimized.