**Exercise 4: Employee Management System**

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

A. **Representation**:

* Arrays are stored in contiguous memory locations, meaning that each element is placed next to the previous one.
* This contiguous allocation allows for constant-time access to elements using an index (i.e., O(1) time complexity).

**Advantages**:

* **Fast Access**: Direct indexing allows for quick retrieval and updating of elements.
* **Memory Efficiency**: Fixed size of arrays means there is no overhead for storing metadata.
* **Simple Implementation**: Easy to implement and use due to straightforward indexing.

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

A. **Add**:

* **Complexity**: O(n) for adding an element at the end if the array is full and needs resizing. O(1) for adding at a specific index if the array has enough capacity.

**Search**:

* **Complexity**: O(n) for searching an unsorted array using linear search. O(log n) if the array is sorted and binary search is used.

**Traverse**:

* **Complexity**: O(n), as you need to visit each element in the array.

**Delete**:

* **Complexity**: O(n) for deleting an element, as elements may need to be shifted to fill the gap. O(n) for deleting from a specific index if elements need to be shifted.

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

**A. Limitations**:

* **Fixed Size**: Arrays have a fixed size once created. They cannot be resized dynamically without creating a new array.
* **Inefficient Insertion/Deletion**: Adding or removing elements (especially from the middle) requires shifting elements, which is inefficient.
* **Wasted Space**: If the array is larger than needed, it may waste memory.

**When to Use**:

* **When Size is Known**: If the number of elements is known beforehand and remains relatively stable.
* **Fast Access Required**: When you need fast, constant-time access to elements.
* **Memory Efficiency**: When memory overhead and additional features of other data structures are not needed.