Employee Management System

Understanding Array Representation

Memory Representation of Arrays:

- **Contiguous Memory Allocation**: Arrays are stored in contiguous blocks of memory. Each element is placed next to its neighbors, allowing for direct access to any element using its index.
- **Fixed Size**: Arrays have a fixed size, determined at the time of allocation. The size cannot be changed once the array is created.
- **Indexing**: Accessing an element in an array is done via indexing, which provides constant-time access (O(1)) to any element.

Advantages of Arrays:

- **Fast Access**: Direct access to any element using its index.
- **Simple Data Structure**: Easy to understand and implement.
- **Memory Efficiency**: Low overhead compared to other data structures like linked lists.

Analysis

Time Complexity Analysis:

- **Add**: O(1) - Adding an employee at the end of the array is constant time, assuming there is space.
- **Search**: O(n) - Searching for an employee by ID requires a linear scan of the array.
- **Traverse**: O(n) - Traversing all employees requires visiting each element once.
- **Delete**: O(n) - Deleting an employee requires shifting elements, which takes linear time in the worst case.

Limitations of Arrays:

1. **Fixed Size**
2. **Inefficient Insertions/Deletions**.
3. **Memory Waste**

When to Use Arrays:

- **When the number of elements is known and fixed**.
- **When constant-time access to elements is required**.
- **When memory overhead needs to be minimized**.

For dynamic and efficient management of employee records, other data structures like ArrayLists, LinkedLists, or HashMaps can be considered, depending on specific requirements such as dynamic resizing, fast insertions/deletions, or efficient lookups.