**Exercise 4: Employee Management System**

1. **Understanding Array Representation:**
   * **Q) How are arrays represented in memory and what are their advantages?**
     + **Ans)** In Java, arrays are collections of elements of the same data type, stored contiguously in memory. They are indexed data structures, with indexing starting from 0. Elements in an array are accessed using their index.

**Advantages:**

* + - **Fast Access**: Accessing an element by its index is very quick, with a time complexity of O(1).
    - **Low Memory Overhead**: Arrays have minimal additional memory overhead since data is stored in contiguous memory locations.
    - **Easy Traversal**: Iterating over an array is straightforward and efficient.

1. **Analysis:**
   * **Q) What is the time complexity of each operation (add, search, traverse, delete) in an array?**
     + **Ans)**
       - **Add**: O(1) — Adding an employee to the array takes constant time, provided there is space available in the array.
       - **Search**: O(n) — In the worst case, the entire array might need to be searched.
       - **Traverse**: O(n) — Traversing through all elements of the array.
       - **Delete**: O(n) — In the worst case, the entire array might need to be iterated through to find the element to delete, followed by shifting the remaining elements.
   * **Q) What are the limitations of arrays and when should they be used?**
     + **Ans)**
       - **Fixed Size**: The size of an array is fixed upon declaration and cannot be changed. This can lead to memory wastage if the array is not fully utilized, and a new array must be created when it becomes full.
       - **Inefficient Deletion and Insertion**: Deleting and inserting elements, especially in the middle of the array, is inefficient as it may require shifting elements.
       - **Better Alternatives**: For dynamic data where the number of elements may change, other data structures like ArrayLists or LinkedLists, or other collections, may be more suitable.