**Exercise 4: Employee Management System**

**Scenario:**

You are developing an employee management system for a company. Efficiently managing employee records is crucial.

**Steps:**

1. **Understand Array Representation:**
   * Explain how arrays are represented in memory and their advantages.
2. **Setup:**
   * Create a class Employee with attributes like **employeeId**, **name**, **position**, and **salary**.
3. **Implementation:**
   * Use an array to store employee records.
   * Implement methods to **add**, **search**, **traverse**, and **delete** employees in the array.
4. **Analysis:**
   * Analyze the time complexity of each operation (add, search, traverse, delete).
   * Discuss the limitations of arrays and when to use them.

**ANSWER:**

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

Arrays are contiguous blocks of memory where each element is of the same type and size. The array's name serves as a reference to the first element, and the elements are indexed starting from 0. The memory address of any element can be calculated using the formula: base\_address + index \* element\_size

**Compare the performance (time complexity) of Bubble Sort and Quick Sort.**

| **Cases** | **Add employee** | **Search Employee** | **Traverse Employee** | **Delete Employee** |
| --- | --- | --- | --- | --- |
| **Worst case** | O(1) | O(n) | O(n) | O(n) |
| **Average case** | - | O(n) | - | O(n) |
| **Best case** | O(1) | O(1) | O(n) | O(1) |

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

* Fixed Size
* Inefficient Insertions and Deletions
* Limited Flexibility