**Employee Management System**

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

**Arrays** are a **collection of elements** stored **at contiguous memory locations**. Each element in an array can be accessed using its index, starting from 0.

* **Advantages of Arrays**:
* **Direct Access**: Accessing an element using its index is very fast, with a time complexity of O(1).
* **Memory Efficiency**: Arrays store elements in contiguous memory locations, which helps in efficient memory usage and quick access.

If you created the array locally it will be on stack. Where the elements are stored depends on the storage specification.

For example:

* A **local array** will be (usually) created on **stack** while
* A **global or static array** will be (usually) created on **bss/data segments** and
* A **dynamically created array** will be created on **heap.**

**Time Complexity Analysis:**

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

* **Add Employee (O(1))**

Adding a new employee to the system takes constant time, regardless of the number of existing employees. This is because the new employee is simply appended to the end of the employee array.

* **Search Employee (O(n))**

Searching for an employee based on their ID involves iterating through the entire employee array in the worst case resulting in linear time complexity.

* **Traverse Employees (O(n))**

Printing the details of all employees requires visiting each employee in the array leading to linear time complexity.

* **Delete Employee (O(n))**

Deleting an employee involves searching for the employee (which can take up to O(n) time) and then shifting the remaining employees in the array (also O(n) in the worst case). Therefore, the overall time complexity is linear.

1. **Discuss why Quick Sort is generally preferred over Bubble Sort.**

Though **Bubble Sort** is easy to understand and implement but inefficient for large lists due to its O(n^2) time complexity. Whereas, **Quick Sort** is more complex but generally much faster due to its O(n log n) average time complexity.

That’s why **Quick Sort is preferred over** **Bubble Sort** for larger datasets.