1. **Understand Array Representation:**

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

SOLN:

An array is a data structure that consists of a collection of elements, each identified by at least one array index or key. Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

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

**Add Operation**

* **Description**: Adding an employee to the array.
* **Time Complexity**:
  + **Best Case**: O(1) - If there is available space in the array.
  + **Worst Case**: O(n) - If the array is full and needs to be resized (requires copying all elements to a new array).

**Search Operation**

* **Description**: Searching for an employee by ID.
* **Time Complexity**:
  + **Worst Case**: O(n) - The algorithm may need to scan through all elements in the array to find the target employee or determine that it is not present.

**Traverse Operation**

* **Description**: Displaying all employees.
* **Time Complexity**: O(n) - Every element in the array must be accessed and printed.

**Delete Operation**

* **Description**: Deleting an employee by ID.
* **Time Complexity**:
  + **Worst Case**: O(n) - The algorithm needs to find the target employee and then shift all subsequent elements to fill the gap left by the deleted element.
* Discuss the limitations of arrays and when to use them.

The limitations are fixed size, wasted memory, Insufficient Insertion and deletion.