Answers

1.Arrays are data structures that store elements of the same type in contiguous memory locations. They provide a way to store and manage multiple items in a single variable, allowing for efficient access and manipulation.

**Advantages of Arrays**

1. **Fast Access**: Arrays allow for constant-time access (O(1)) to elements through their indices, which is useful for operations that require frequent retrieval.
2. **Memory Efficiency**: Arrays have a fixed size, which minimizes overhead, as they do not require additional memory for pointers or references (unlike linked structures).

4. **Analysis**

**Time Complexity of Operations**

1. **Add Employee**: O(1) (amortized) if space is available.
2. **Search Employee**: O(n) (in the worst case, needs to check all employees).
3. **Traverse Employees**: O(n) (since it needs to access each employee).
4. **Delete Employee**: O(n) (in the worst case, needs to check all employees).

**Limitations of Arrays**

1. **Fixed Size**: Arrays have a fixed size, which means that you cannot easily add or remove elements after their creation without creating a new array.
2. **Inefficient Deletion**: Deleting an element requires shifting subsequent elements, leading to O(n) time complexity.
3. **Poor Memory Utilization**: If the array size is not optimally chosen, it may lead to wasted space (if it's too large) or overflow (if it's too small).

**When to Use Arrays**

Arrays are best used when:

* You know the maximum number of elements in advance.
* Fast access to elements by index is required.