**Understanding Array Representation**

**Array Representation in Memory**

In computer science, an array is a data structure that stores a collection of elements, each identified by an index or key. In memory, an array is represented as a contiguous block of memory locations, where each location stores a single element of the array.

**Advantages of Arrays**

Arrays have several advantages:

* **Efficient Memory Usage**: Arrays store elements in contiguous memory locations, making them memory-efficient.
* **Fast Access**: Arrays provide fast access to elements using their index.
* **Simple Implementation**: Arrays are easy to implement and understand.

**Analysis**

**Time Complexity Analysis**

* **Add Employee**: O(1), since we simply add the employee to the end of the array.
* **Search Employee**: O(n), where n is the number of employees, since we need to iterate through the array to find the employee.
* **Traverse Employees**: O(n), where n is the number of employees, since we need to iterate through the array to print all employees.
* **Delete Employee**: O(n), where n is the number of employees, since we need to iterate through the array to find the employee and shift elements to the left.

**Limitations of Arrays**

Arrays have several limitations:

* **Fixed Size**: Arrays have a fixed size, which can lead to memory waste if the array is not fully utilized.
* **Slow Search**: Arrays have a slow search time, especially for large datasets.
* **Slow Insertion and Deletion**: Arrays have slow insertion and deletion times, especially when elements need to be shifted.

**When to Use Arrays**

Arrays are suitable when:

* **Memory Efficiency**: Memory efficiency is crucial, and the dataset is relatively small.
* **Fast Access**: Fast access to elements is required, and the dataset is relatively small.
* **Simple Implementation**: A simple implementation is desired, and the dataset is relatively small.