**Exercise 4: Employee Management System**

**1. Understand Array Representation**

Array Representation in Memory

In computer science, an array is a data structure that consists of a collection of elements (values or variables), each identified by an array index. Arrays are stored in contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easy to calculate the position of each element by simply adding an offset to a base value, i.e., the memory location of the first element of the array (typically denoted by the name of the array).

Advantages of Arrays

1. Random Access: Arrays allow constant-time O(1) access to elements, as you can directly access any element using its index.
2. Memory Efficiency: Arrays are stored in contiguous memory locations, which minimizes the overhead of pointers or other structures, leading to efficient memory usage.
3. Cache Performance: Due to their contiguous storage, arrays benefit from spatial locality, leading to better cache performance.
4. Ease of Use: Arrays are straightforward to declare and use in many programming languages, making them a fundamental data structure.

**4.Analysis**

**1.Time Complexity:**

If there is space in the array, add O(1); if not, handle the entire array in O(n).  
Search: O(n) since a linear search throughout the array is necessary.  
Traverse: O(n), so each element must be visited once.  
Delete: O(n) since elements must be shifted after deletion.

2.Limitations of Arrays:  
  
Fixed Size: An array's size cannot be altered once it has been constructed.  
Ineffective Insertions and Deletions: Because shifting pieces are required, inserting or removing elements, particularly in the middle, can be expensive.

3.When to Apply Arrays:  
  
Arrays are appropriate when index-based access requires excellent performance or when the number of elements is known and fixed. It is better to use another data structure, such as an array list, linked list, or hash map, for dynamic sizes and frequent insertions and deletions.