Exercise 4: Employee Management System

**Time Complexity of Operations:**

* **Add:** O(1) - Adding an element to the array is generally constant time, assuming there is space available.
* **Search:** O(n) - Searching for an element requires a linear scan, making it O(n) in the worst case.
* **Traverse:** O(n) - Traversing all elements takes linear time, O(n).
* **Delete:** O(n) - Deleting an element involves finding it and potentially shifting elements, making it O(n).

**Limitations of Arrays:**

* **Fixed Size:** Arrays have a fixed size, which limits the number of elements they can store. This can lead to either wasted memory (if the array is too large) or insufficient capacity (if the array is too small).
* **Insertion/Deletion Efficiency:** Insertion and deletion can be inefficient, especially if elements need to be shifted. This makes arrays less suitable for scenarios where frequent insertions and deletions are required.
* **Memory Contiguity:** Arrays require contiguous memory allocation, which can be a limitation in systems with fragmented memory.

**When to Use Arrays:**

* Use arrays when the number of elements is known in advance and does not change frequently.
* They are ideal for scenarios requiring fast access to elements by index.