Que1. Explain how arrays are represented in memory and their advantages?

Ans: An array is a collection of same type of elements, stored in continuous memory locations

Advantages:

* Accessing elements takes O(1) time complexity
* Arrays are often used for sorting and searching data.
* Arrays allow for fast data retrieval because the data is stored in contiguous memory locations.
* Arrays can be used to store a wide range of data types, including integers, floating-point numbers, characters, and even complex data structures such as objects

Que2. Analyse the time complexity of each operation (add, search, traverse, delete)?

1. Add: time complexity of this function if O(1) as it adds element to the last
2. Search: time complexity of this function is O(n) because I used linear search but if used binary search it would have been O(logn)
3. Traverse: traversing every element in an array would take a time complexity of O(N)
4. Delete: time complexity of this function is O(n) as after removing an element we have to shift rest of the elements to cover the empty space

Que3. Discuss the limitations of arrays and when to use them?

Ans: Limitations of arrays:

* Once an array is initialized with a fixed size, its capacity cannot be changed. This can lead to either wasted space if the array is too large or the need for resizing if the array is too small.
* Inserting an element into the middle of an array requires shifting all subsequent elements to the right, which takes O(n) time.
* Deleting an element from the middle of an array requires shifting all subsequent elements to the left, which also takes O(n) time.
* Accessing elements outside the array's bounds can lead to runtime errors or undefined behaviour, making bound checking necessary.

When to use arrays:

* When the number of elements is known in advance and does not change, arrays are suitable due to their fixed size and predictable memory usage.
* Arrays provide O(1) time complexity for accessing elements using their index, making them ideal for applications requiring frequent and fast random access.
* When the collection of elements does not change over time, arrays provide a reliable and efficient way to store and access data.