**Ans1.Difference between time complexity of Array and link list.**

**Dynamic Array:-**

a .Array takes more time while performing any operation like insertion, deletion, etc.

b.Accessing any element in an array is faster as the element in an array can be directly accessed through the index.

c.In a fixed-length array, appends only take O(1) time. But appends take O(n) time only when we insert into a full array.

**Linked List:-**

a.Linked list takes less time while performing any operation like insertion, deletion, etc.

b.Accessing an element in a linked list is slower as it starts traversing from the first element of the linked list.

c.For insertion in the linked list, the time complexity is O(1) if done on the head, O(N) if done at any other location.

**Part2:- Space Complexity**

**Dynamic Array:-**

The elements of the dynamic array are stored contiguously at the start of the underlying array, and the remaining positions towards the end of the underlying array are reserved, or unused. Elements can be added at the end of a dynamic array in constant time by using the reserved space until this space is completely consumed. When all space is consumed, and an additional element is to be added, the underlying fixed-sized array needs to be increased in size.

**Linked list:-**

Space complexity for both types of linked lists is O(n), as each element requires space for its data and pointers to the next (and possibly previous) node, resulting in linear space usage proportional to the number of elements.

**Part 3 :-**

**Advantages Of Linked List:**

**Dynamic data structure:** A linked list is a dynamic arrangement so we can increase and decrease it at runtime by allocating and deallocating memory. So there is no need to give the initial size of the linked list.

**No memory wastage:** In the Linked list, efficient memory utilization can be achieved since the size of the linked list increase or decrease at run time so there is no memory wastage and there is no need to pre-allocate the memory.

**Implementation:** Linear data structures like stacks and queues are often easily implemented using a linked list.

Insertion and Deletion Operations: Insertion and deletion operations are quite easier in the linked list. There is no need to shift elements after the insertion or deletion of an element only the address present in the next pointer needs to be updated.

**Flexible:** This is because the elements in Linked List  are not stored in contiguous memory locations unlike the array.

**Disadvantages Of Linked List:**

**Memory usage:** More memory is required in the linked list as compared to an array. Because in a linked list, a pointer is also required to store the address of the next element and it requires extra memory for itself.
**Traversal:** In a Linked list traversal is more time-consuming as compared to an array. Direct access to an element is not possible in a linked list as in an array by index. For example, for accessing a node at position n, one has to traverse all the nodes before it.
**Reverse Traversing:** In a singly linked list reverse traversing is not possible, but in the case of a doubly-linked list, it can be possible as it contains a pointer to the previously connected nodes with each node. For performing this extra memory is required for the back pointer hence, there is a wastage of memory.
**Random Access:** Random access is not possible in a linked list due to its dynamic memory allocation.

**Advantages of Arrays :**
Below are some advantages of the array:

1.Dynamic arrays are expandable: A dynamic array can increase in size as needed, which makes it especially useful when you don't know how large the array will be. This way, you can avoid the issue of having a static array that is too small to store the data you need to store.
2.Dynamic arrays are efficient: When compared to linked lists and hash tables, a dynamic array can be more efficient when you need to retrieve elements quickly. Since a dynamic array is indexed, all elements can be accessed and retrieved in constant time (O(1)). In contrast, with a linked list, it can take O(n) time, where n is the number of elements in the list.

**Disadvantages of a Dynamic Array:**
Like with any data structure, there are some drawbacks to using a dynamic array. Here are some of the potential drawbacks:

1.Dynamic arrays can be slow to resize: When the number of elements in a dynamic array increases and the array needs to be resized, the entire array must be moved to a new location in memory, which can be costly in terms of time and resources.
2.Dynamic arrays can use a lot of memory: A dynamic array often needs to allocate more memory than is strictly necessary, since it needs to allocate extra space in order to be able to store more elements when needed. This can result in memory wastage.