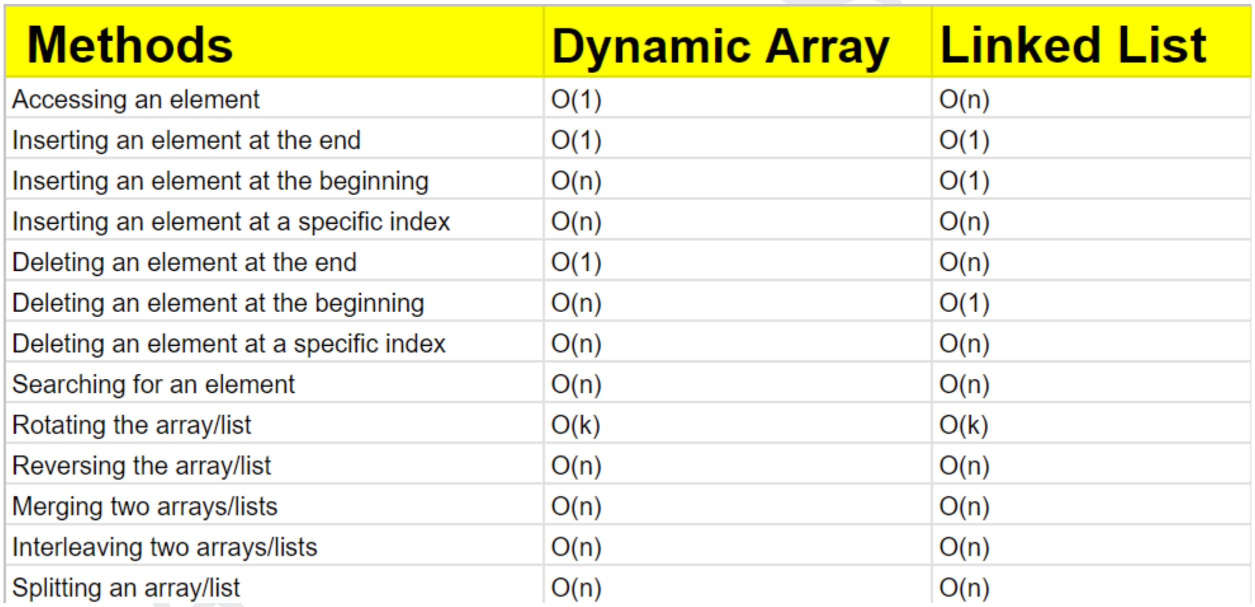
SEMESTER-02

-Abhishek Singh (2301010006)

Comparison of Linked Lists and Dynamic Arrays-

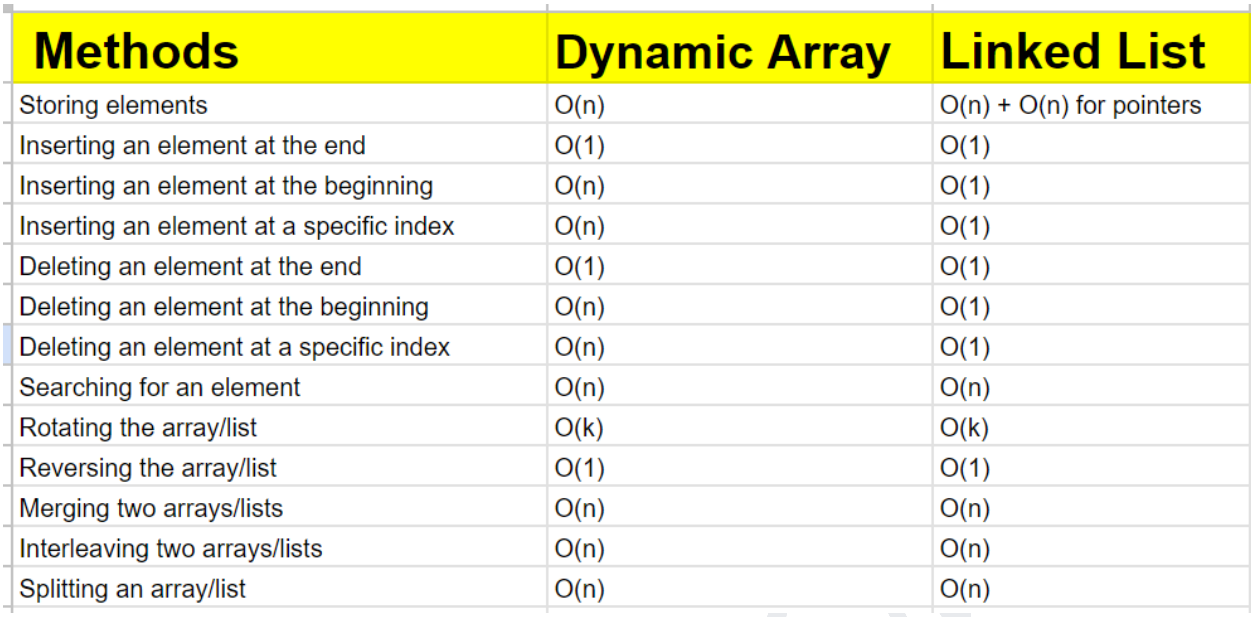
• Compare the following aspects of linked lists and dynamic arrays:

1>>>> Time complexity of each method :-



Methods	Dynamic Array	Linked List
Accessing an element	O(1)	O(n)
Inserting an element at the end	O(1)	O(1)
Inserting an element at the beginning	O(n)	O(1)
Inserting an element at a specific index	O(n)	O(n)
Deleting an element at the end	O(1)	O(n)
Deleting an element at the beginning	O(n)	O(1)
Deleting an element at a specific index	O(n)	O(n)
Searching for an element	O(n)	O(n)
Rotating the array/list	O(k)	O(k)
Reversing the array/list	O(n)	O(n)
Merging two arrays/lists	O(n)	O(n)
Interleaving two arrays/lists	O(n)	O(n)
Splitting an array/list	O(n)	O(n)

2>> Space complexity of each method :-



Methods	Dynamic Array	Linked List
Storing elements	O(n)	O(n) + O(n) for pointers
Inserting an element at the end	O(1)	O(1)
Inserting an element at the beginning	O(n)	O(1)
Inserting an element at a specific index	O(n)	O(1)
Deleting an element at the end	O(1)	O(1)
Deleting an element at the beginning	O(n)	O(1)
Deleting an element at a specific index	O(n)	O(1)
Searching for an element	O(n)	O(n)
Rotating the array/list	O(k)	O(k)
Reversing the array/list	O(1)	O(1)
Merging two arrays/lists	O(n)	O(n)
Interleaving two arrays/lists	O(n)	O(n)
Splitting an array/list	O(n)	O(n)

Advantages and disadvantages of each data structure

Dynamic Array:

Advantages:

1. Dynamic arrays can grow and shrink dynamically, so there is no need to pre-allocate memory.
2. Accessing an element in a dynamic array is faster than in a linked list.
3. Dynamic arrays are more space-efficient than linked lists.

Disadvantages:

1. Inserting or deleting an element at the beginning or in the middle of a dynamic array can be expensive, as all subsequent elements need to be shifted.
2. Dynamic arrays have a fixed capacity, so if the array is full, it needs to be resized, which can be expensive.

Linked List:

Advantages:

1. Linked lists are more flexible than dynamic arrays because elements are not stored in contiguous memory locations.
2. Inserting or deleting an element at the beginning or in the middle of a linked list is faster than in a dynamic array.
3. Linked lists can handle large datasets that can grow and shrink dynamically.

Disadvantages:

1. Accessing an element in a linked list is slower than in a dynamic array.
2. Linked lists require more memory than dynamic arrays because of the need for pointers.
3. Traversing a linked list is slower than in a dynamic array.

• Create a report on comparison of both of them

- Dynamic arrays and linked lists are two popular data structures used in programming. Dynamic arrays are more space-efficient than linked lists, and accessing an element in a dynamic array is faster than in a linked list.

- However, inserting or deleting an element at the beginning or in the middle of a dynamic array can be expensive, as all subsequent elements need to be shifted.

- Dynamic arrays have a fixed capacity, so if the array is full, it needs to be resized, which can be expensive. Linked lists, on the other hand, are more flexible