

**Name - Harshit Raj**

**Enrollment No. - 2301010046**

## **Comparison of Linked Lists and Dynamic Arrays**

### **1. Time Complexity**

<b>Operations</b>	<b>Linked List</b>	<b>Dynamic Array</b>
Get size	$O(1)$	$O(1)$
Is Empty	$O(1)$	$O(1)$
Append	$O(n)$	$O(1)$ (amortized)
Prepend	$O(1)$	$O(n)$
Insert at index	$O(n)$	$O(n)$
Delete at index	$O(n)$	$O(n)$
Reverse	$O(n)$	$O(n)$
Rotate right by k	$O(n)$	$O(n)$
Find middle	$O(n)$	$O(1)$
Index of element	$O(n)$	$O(n)$
Merge	—	$O(n+m)$
Interleave	—	$O(n+m)$
Split at index	—	$O(n)$
Resize (custom factor)	—	$O(1)$

## 2. Space Complexity

Operations	Linked List	Dynamic Array
Get size	$O(1)$	$O(1)$
Is Empty	$O(1)$	$O(1)$
Append	$O(1)$	$O(1)$ (amortized)
Prepend	$O(1)$	$O(n)$
Insert at index	$O(1)$	$O(n)$
Delete at index	$O(1)$	$O(n)$
Reverse	$O(1)$	$O(1)$
Rotate right by k	$O(1)$	$O(1)$
Find middle	$O(1)$	$O(1)$
Index of element	$O(1)$	$O(1)$
Merge	—	$O(n+m)$
Interleave	—	$O(n+m)$
Split at index	—	$O(n)$
Resize (custom factor)	—	$O(1)$

## 3. Advantages and disadvantages of each data structure

### Linked List

#### Advantages :

1. Dynamic size
2. Efficient insertions and deletion
3. No space wastage

**Disadvantage :**

1. Slow access
2. Memory overhead
3. Cache performance

## **Dynamic Arrays**

**Advantages :**

1. Fast access
2. Efficient iteration
3. Memory efficiency

**Disadvantage :**

1. Resize overhead
2. Insert and delete costs
3. Pre-allocated space