# Name - Priya Saha

Enrollment No. - 2301010085

## Comparison of Linked Lists and Dynamic Arrays

### 1. Time Complexity

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

## 2. Space Complexity

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

## 3. Advantages and disadvantages of each data structure

### Linked List

| Advantages | Disadvantage |
| --- | --- |
| 1. Dynamic size | 1. Slow access |
| 2. Efficient insertions and deletion | 2. Memory overhead |
| 3. No space wastage | 3. Cache performance |

### Dynamic Arrays

| Advantages : | Disadvantage : |
| --- | --- |
| 1. Fast access | 1. Resize overhead |
| 2. Efficient iteration | 2. Pre-allocated space |
| 3. Memory efficiency | 3. Insert and delete costs |