Q3) assignment

Time and Space complexity of all methods that I have used in the Q1.

| **Method** | **Time Complexity** | **Space Complexity** |
| --- | --- | --- |
| insertAt | O(n) | O(1) |
| deleteAt | O(n) | O(1) |
| size | O(n) | O(1) |
| isEmpty | O(1) | O(1) |
| rotateRight | O(n) | O(1) |
| reverse | O(n) | O(1) |
| append | O(n) | O(1) |
| prepend | O(1) | O(1) |
| merge | O(n) | O(1) |
| interleave | O(n) | O(1) |
| middleElement | O(n) | O(1) |
| indexOf | O(n) | O(1) |
| splitAt | O(n) | O(1) |
| printList | O(n) | O(1) |

Time and Space complexity of all methods that I have used in the Q2.

| **Method** | **Time Complexity** | **Space Complexity** |
| --- | --- | --- |
| add | O(1)\* | O(1) |
| insertAt | O(n) | O(n) |
| removeAt | O(n) | O(1) |
| get | O(1) | O(1) |
| set | O(1) | O(1) |
| size | O(1) | O(1) |
| isEmpty | O(1) | O(1) |
| ensureCapacity | O(n)\* | O(n) |
| printArray | O(n) | O(1) |

| **Aspect** | **Dynamic Array** | **Linked List** |
| --- | --- | --- |
| Advantages | - Random access allowed  - Memory efficiency  - Cache locality  - Efficient appends  - Easy to handle | - Dynamic size  - Efficient insertions and deletions  - Do not need Contiguous memory location  - No resizing overhead |
| Disadvantages | - Costly resizing  - Fixed size  - Need a contiguous memory location  - Costly insertions and deletions | - No random access  - Poor cache locality  - Inefficient appends  - It takes extra space to store next node pointer  - Not easy to handle |