**Q: Explain the different types of linked lists (Singly Linked List, Doubly Linked List).**

**Understanding Linked Lists:** A linked list is a linear data structure where elements are not stored in contiguous memory locations. Instead, each element, known as a node, contains data and a reference (link) to the next node in the sequence.

**Types of Linked Lists:**

* **Singly Linked List:** Each node has a reference to the next node in the list, allowing traversal in only one direction.
* **Doubly Linked List:** Each node has references to both the next and the previous nodes, enabling traversal in both directions.

**Q: Analyze the time complexity of each operation.**

**Time Complexity Analysis:**

* **Add:**
  + **Worst Case:** O(n) when adding at the end of the list.
  + **Best Case:** O(1) when adding at the beginning.
* **Search:** O(n) - Requires scanning through the list to find the desired element.
* **Traverse:** O(n) - Involves visiting each node in the list.
* **Delete:**
  + **Worst Case:** O(n) when deleting the last element or an element in the middle.

**Q: Discuss the advantages of linked lists over arrays for dynamic data.**

**Advantages of Linked Lists over Arrays:**

* **Dynamic Size:** Linked lists can dynamically grow or shrink, unlike arrays which have a fixed size determined at creation.
* **Efficient Insertions and Deletions:** Adding or removing elements, especially in the middle of the list, is generally faster in a linked list compared to an array due to the lack of need for shifting elements.
* **Flexibility:** Linked lists are versatile and can be used to implement various other data structures such as stacks, queues, and graphs.