**Week-I**

**Algorithms & DSA**

**Exercise-5:**

**Types of Linked Lists:**

1. **Singly Linked List**:
   * **Structure**: Each node contains data and a reference to the next node.
   * **Traversal**: Can be done in one direction (from the head to the end).
   * **Operations**: Insertion and deletion can be efficient if the position is known.
2. **Doubly Linked List**:
   * **Structure**: Each node contains data, a reference to the next node, and a reference to the previous node.
   * **Traversal**: Can be done in both directions (forward and backward).
   * **Operations**: More flexible and efficient for certain operations like deletion of a given node.

**Time Complexity Analysis:**

* **Add Operation**: O(n) - Adding a task at the end of the list requires traversal to the end.
* **Search Operation**: O(n) - Searching requires scanning through the list.
* **Traverse Operation**: O(n) - Traversing all elements in the list takes linear time.
* **Delete Operation**: O(n) - Deleting requires searching for the element and updating references, which takes linear time.

**Advantages of Linked Lists Over Arrays for Dynamic Data:**

**Dynamic Size**: Linked lists can grow and shrink dynamically, unlike arrays which have a fixed size.

**Efficient Insertions/Deletions**: Insertions and deletions are more efficient in linked lists, especially when dealing with large amounts of data, since they do not require shifting elements.

**Memory Utilization**: Linked lists can be more memory efficient for dynamic data as they do not require a contiguous block of memory.