**1. Understand Linked Lists**

**Singly Linked List**

* **Description:** A singly linked list is a collection of nodes where each node contains data and a reference to the next node in the sequence. The last node points to null.
* **Advantages:**
  + **Dynamic Size:** Can grow and shrink dynamically as needed.
  + **Efficient Insertions/Deletions:** Insertions and deletions are more efficient than arrays since there is no need to shift elements.

**Doubly Linked List**

* **Description:** A doubly linked list is similar to a singly linked list but each node contains a reference to both the next node and the previous node.
* **Advantages:**
  + **Bi-directional Traversal:** Allows traversal in both directions (forward and backward).
  + **Efficient Deletions:** Deletions are more efficient as there is no need to traverse the list to update the previous node.

**4. Analysis**

**Time Complexity of Each Operation**

* **Add Task:**
  + Time Complexity: O(n) (since we need to traverse to the end of the list)
* **Search Task:**
  + Time Complexity: O(n) (since we may need to traverse the entire list)
* **Traverse Tasks:**
  + Time Complexity: O(n)
* **Delete Task:**
  + Time Complexity: O(n) (since we may need to traverse the entire list)

**Advantages of Linked Lists Over Arrays**

* **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, especially when dealing with large data sets, as there is no need to shift elements as in arrays.