**TASK MANAGEMENT SYSTEM**

**Singly Linked List**

* **Description**: A data structure where each element (node) contains a value and a reference (link) to the next node in the sequence.
* **Advantages**:
  + Dynamic size: Can grow and shrink as needed.
  + Efficient insertions and deletions: Can insert or delete nodes without shifting elements.
* **Disadvantages**:
  + No direct access to elements: Must traverse the list to find an element.

**Doubly Linked List**

* **Description**: A linked list where each node contains a value, a reference to the next node, and a reference to the previous node.
* **Advantages**:
  + Bidirectional traversal: Can traverse the list in both directions.
  + Efficient insertions and deletions: Can insert or delete nodes without shifting elements.
* **Disadvantages**:
  + Extra memory usage: Requires additional memory for the previous node reference.

**Time Complexity of Each Operation**

* **Add**: O(n) - Adding a task involves traversing to the end of the list.
* **Search**: O(n) - Searching for a task requires traversing the list.
* **Traverse**: O(n) - Traversing the list involves visiting each node.
* **Delete**: O(n) - Deleting a task involves searching for the task and updating links.

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

* **Dynamic Size**: Linked lists can grow and shrink dynamically, whereas arrays have a fixed size.
* **Efficient Insertions and Deletions**: Insertions and deletions in linked lists are more efficient than in arrays because they do not require shifting elements.
* **Memory Utilization**: Linked lists allocate memory for each element separately, which can be more efficient when dealing with an unknown or varying number of elements.