

## **TASK MANAGEMENT SYSTEM**

### **Q1) Understand Linked Lists:**

A linked list is a data structure in which a sequence of nodes are linked together through pointers. Each node in the list contains a value and a reference (i.e., a "link") to the next node in the list. This structure allows for efficient insertion and deletion of nodes at any position in the list.

Types of Linked Lists:

#### **1. Singly Linked List:**

In a singly linked list, each node only has a reference to the next node in the list. Each node typically consists of two parts: the data

and the reference to the next node. This means that each node only points to the next node, but not to the previous node.

#### **2. Doubly Linked List:**

In a doubly linked list, each node has references to both the next node and the previous node in the list. Each node consists of three parts : the data,

the reference to the previous node and the reference to the next node. This allows for efficient traversal in both forward and backward directions.

#### **3. Circularly Linked List:**

In a circularly linked list, the last node points back to the first node, forming a circle. This allows for efficient traversal in a circular fashion.

### **Q4) Time complexity analysis:**

addTask:  $O(n)$  where  $n$  is the number of tasks in the list

searchTask:  $O(n)$  where  $n$  is the number of tasks in the list

traverseTasks:  $O(n)$  where  $n$  is the number of tasks in the list

deleteTask:  $O(n)$  where  $n$  is the number of tasks in the list

### **Advantages of linked lists over arrays for dynamic data:**

Linked list allow for efficient insertion and deletion of elements at any position, without requiring shifting of elements.

This makes them ideal for applications where data is frequently added or removed. Secondly, linked lists can allocate

memory dynamically, which is useful when the amount of data is unknown or variable. Additionally, linked lists are suitable

for sparse data, where most elements are empty or null, as they only allocate memory for used nodes. Overall, linked lists

provide a flexible and efficient data structure for dynamic data, making them a popular choice in many applications.