**Understanding Linked Lists**

A linked list is a linear data structure where elements, called *nodes*, are connected using pointers. Each node stores the data and a reference to the next node. Linked lists are particularly handy when you're dealing with data that can grow or shrink frequently during the execution of a program.

**✦ Types of Linked Lists:**

* **Singly Linked List:**In this structure, each node holds a value and a reference to the next node. It allows navigation in a single direction—from the first node (head) to the last.
* **Doubly Linked List:**Here, each node keeps track of both its next and previous neighbors. This makes it easier to traverse forwards and backwards but at the cost of using extra memory for the additional pointer.

**Performance Analysis of Operations**

**1. Add Task:**

* *O(1)* if added at the beginning
* *O(n)* if added at the end without a tail reference

**2. Search Task:**

* *O(n)* since each node might need to be checked

**3. Traverse Tasks:**

* *O(n)* because all nodes must be visited

**4. Delete Task:**

* *O(n)* as you need to find the task and update links

**Why Linked Lists Are Good for Dynamic Data**

* **Flexible Size**: Can grow or shrink easily without fixed limits.
* **Quick Insert/Delete**: No need to shift elements like in arrays.
* **Memory Efficient**: Uses memory only when needed, reducing waste.