

## Ex 1: Inventory Management System

- They help store and manage large inventories efficiently.
- Enable fast search, update, and deletion of products.
- Ensure the system is scalable and doesn't slow down as data grows.
- Improve memory usage and overall performance.

### Types of data structures

- **ArrayList:**  
Ideal for **simple, small inventories** where maintaining **insertion order** is important.
- **HashMap:**  
Best for **fast search, update, and delete** operations using **product ID** as the key.
- **TreeMap:**  
Useful when you need **products sorted by ID** or other keys.
- **LinkedList:**  
Suitable when you frequently **insert or delete items at the beginning or end**.

### Time Complexity:

#### Add:

- Directly appending to the end of the list is fast and constant time
- Time Complexity:  $O(1)$

#### Update:

- You need to search through the list to find the product by productid
- Time Complexity:  $O(n)$

#### Delete:

- Requires a linear search to find the product and then shifts elements after deletion
- Time Complexity:  $O(n)$

### How to Optimize These Operations

- **Minimize search:** When possible, **avoid repeated searches** by caching or combining operations.
- **Batch updates:** Group similar updates or deletions to reduce iteration overhead.