**Data structures and algorithms are essential in handling large inventories-:**

The system must support operations like:

* Adding new items when stock arrives
* Updating quantities after sales or transfers
* Deleting items that are discontinued or expired
* Fetching records instantly for processing orders

Without well-structured data handling, each of these operations could slow down significantly as the inventory scales. Algorithms and data structures help us perform these actions in **predictable, optimized time**.

**The types of data structures suitable-:**

### **ArrayList –** Simple and Suitable for Small Inventories

For small-scale inventory systems—like managing less than a few hundred products—an ArrayList can be a reasonable starting point. It allows you to store product objects in a simple, ordered list and works well when the product list is not too large or frequently modified.

List<Product> inventory = new ArrayList<>();

## **HashMap:**Optimization for Large Inventories

As the warehouse grows and starts handling thousands of items, performance becomes a priority. That’s where HashMap shines.

Map<String, Product> inventory = new HashMap<>();

Here, each product is stored using a **unique key**—like a SKU or barcode.

### **Time Complexity of Operations in ArrayList:**

| **Operation** | **Time Complexity** | **Reason** |
| --- | --- | --- |
| **Add (at end)** | O(1) amortized | Appending at the end is efficient unless resizing is needed |
|  |  |  |
| **Update (by index)** | O(1) | Direct access using index |
|  |  |  |
| **Delete (by ID)** | O(n) | Need to search for ID and then shift remaining elements |

### **Time Complexity in HashMap:**

| **Operation** | **Time Complexity** | **Reason** |
| --- | --- | --- |
| **Add** | O(1) average | Uses hash function to locate bucket quickly |
| **Update** | O(1) average | Direct access via key |
| **Delete (by ID)** | O(1) average | Fast removal using key |
|  |  |  |

## Worst-case time can degrade to O(n) only in extreme cases like many hash collisions—but with good hash functions, this is rare.

### **We can use HashMap for optimization**. HashMap is a Better Choice for Large Inventories because:

## Instant lookup: No need to iterate through a list

## Scalability: Works efficiently even with millions of records

## Cleaner Code: Accessing by inventory.get("PRD-102") is much more readable and direct

## 