**1. Understanding the Problem**

**Description:  
• Managing a large inventory involves storing, searching, updating, and deleting product records quickly and efficiently.  
• Data structures and algorithms are essential because they impact speed, memory usage, and scalability of the system.  
• A poor choice can lead to slow lookups, redundant storage, and inefficient updates in a warehouse with thousands of items.**

**Suitable Data Structures:**

| **Data Structure** | **Description** |
| --- | --- |
| **Array/List** | **Simple, but inefficient for large data (slow search/update).** |
| **Hash Map (Dictionary)** | **Best for fast lookup, insert, and delete by unique key like productId.** |
| **Binary Search Tree (BST)** | **Good for sorted access but slower than HashMap for direct key access.** |
| **Heap** | **Useful for finding highest or lowest stock levels quickly.** |
| **Queue/Stack** | **Useful for order processing or inventory history, not for direct lookup.** |

**Recommended: Hash Map – for efficient access using product ID.**

**4. Analysis**

**Time Complexity (Using Hash Map)**

| **Operation** | **Description** | **Time Complexity** |
| --- | --- | --- |
| **Add** | **Add a new product using product ID** | **O(1) average** |
| **Update** | **Update stock/price by product ID** | **O(1) average** |
| **Delete** | **Remove a product by product ID** | **O(1) average** |
| **Search** | **Retrieve details by product ID** | **O(1) average** |

**Note: Worst-case time complexity can degrade to O(n) in poor hash function scenarios, but proper hash design avoids this.**

**Optimization Techniques**

| **Operation** | **Optimization Strategy** |
| --- | --- |
| **Add** | **Use unique productId as key in hash table for fast insertion.** |
| **Update** | **Direct access via key avoids search time. Keep update operations atomic.** |
| **Delete** | **Simple key-based deletion in hash table. Periodically clean up unused space.** |
| **Search** | **Use hash index or secondary index for multi-attribute search (e.g., name, category).** |

**When to Use What**

| **Situation** | **Recommended Structure** |
| --- | --- |
| **Fast access by product ID** | **Hash Map** |
| **Need to maintain sorted order** | **Tree (e.g., AVL Tree)** |
| **Need to track stock min/max** | **Heap** |
| **History of inventory changes** | **Stack or Queue** |