**1. Understand the Problem**

**Why data structures and algorithms are essential in handling large inventories:**

* Efficient data storage and retrieval are vital to ensure quick access to inventory details, minimizing delays in processing transactions and managing stock levels.
* Proper algorithms and data structures allow for efficient searching, insertion, deletion, and updating of inventory items, which is crucial for maintaining the accuracy and performance of the system as the inventory grows.

**Types of data structures suitable for this problem:**

* **ArrayList**: Provides dynamic array capabilities, allowing for easy addition and retrieval of items. However, it has slower deletion and update operations as elements need to be shifted.
* **HashMap**: Offers average constant-time complexity (O(1)) for insertion, deletion, and search operations. It is suitable for key-value pairs, such as product ID to product details, making it an ideal choice for this problem.

**2. Setup**

Create a new project for the inventory management system in your preferred development environment (e.g., Visual Studio Code).

**3. Implementation**

Let's implement the code using Java. We'll define a Product class and use a HashMap to store the products. We'll also implement methods to add, update, and delete products.

**4. Analysis**

**Time Complexity:**

* **Add Product (addProduct)**: Average O(1) - Constant time complexity as HashMap allows average O(1) insertion.
* **Update Product (updateProduct)**: Average O(1) - Constant time complexity for getting the product and updating its attributes.
* **Delete Product (deleteProduct)**: Average O(1) - Constant time complexity for removing an item from HashMap.

**Optimization:**

* Using HashMap already optimizes these operations with average O(1) complexity.
* Further optimization can be achieved by ensuring the HashMap has an appropriate initial capacity and load factor to minimize rehashing.

**Running the Code**

To run this code in Visual Studio Code:

1. Create a new Java project.
2. Create Product.java and InventoryManagementSystem.java files with the provided code.
3. Open the terminal in Visual Studio Code.
4. Compile the Java files using javac Product.java InventoryManagementSystem.java.
5. Run the compiled classes using java InventoryManagementSystem.

This should provide you with a functional inventory management system that can add, update, and delete products efficiently.