**INVENTORY MANAGEMENT SYSTEM:**

**1. Understanding the Problem Requirements**

* **Efficiency:** Proper data structures ensure efficient storage, retrieval, and manipulation of data. In an inventory system, operations such as adding, updating, and deleting products need to be fast and efficient.
* **Scalability:** Efficient algorithms and data structures allow the system to handle a growing amount of data without significant performance degradation.
* **Data Integrity:** Using appropriate data structures helps maintain the integrity and consistency of the inventory data.

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

* **ArrayList:** Provides fast random access and is suitable for scenarios where the size of the inventory changes dynamically.
* **HashMap:** Provides average O(1) time complexity for insertion, deletion, and lookup operations, making it ideal for large inventories where quick access to product details is essential.

**2. Setup**

Creating a new Java project named InventoryManagementSystem.

**3. Implementation:** A class Product with attributes like productId, productName, quantity, and price

**Choosing an appropriate data structure to store the products:**

For this implementation, we will use a HashMap to store products, where the key is the productId and the value is the Product object.

**Implementing methods to add, update, and delete products from the inventory:**

* addProduct(Product product): Adds a new product to the inventory.
* updateProduct(String productId, Product updatedProduct): Updates the details of an existing product.
* deleteProduct(String productId): Removes a product from the inventory.
* getProduct(String productId): Retrieves a product from the inventory.

**4. Analysis**

**Time complexity of each operation in the HashMap data structure:**

* **Add Product (addProduct):** O(1) on average.
* **Update Product (updateProduct):** O(1) on average.
* **Delete Product (deleteProduct):** O(1) on average.
* **Get Product (getProduct):** O(1) on average.

**Optimization of operations:**

* **Indexing:** Ensure the productId is unique to prevent collisions in the HashMap.
* **Batch Operations:** For bulk updates or deletions, consider batch processing to reduce the overhead of multiple operations.

This solution (code included in the same directory) provides a detailed explanation and implementation of an inventory management system using a HashMap for efficient data storage and retrieval.