**Understanding the Problem**

In an inventory management system, efficient data storage and retrieval are crucial to ensure accurate tracking of products, quantities, and prices. Data structures and algorithms play a vital role in handling large inventories by:

* Enabling fast search, insertion, and deletion of products
* Optimizing storage space and reducing memory usage
* Improving system performance and scalability

For this problem, suitable data structures include:

* **ArrayList**: A dynamic array-based data structure that allows for efficient insertion and deletion of elements.
* **HashMap**: A key-value data structure that enables fast lookup, insertion, and deletion of products using their IDs as keys.

**Time Complexity Analysis**

* **Add Operation**: O(1) average case, since we're using a HashMap with a unique productId as the key.
* **Update Operation**: O(1) average case, since we're updating an existing product in the HashMap.
* **Delete Operation**: O(1) average case, since we're removing a product from the HashMap using its productId.

**Optimization**

To further optimize these operations, we can:

* Use a more efficient data structure, such as a balanced binary search tree (e.g., AVL tree, Red-Black tree), which would provide O(log n) time complexity for search, insertion, and deletion operations.
* Implement caching to reduce the number of database queries or disk I/O operations.
* Use parallel processing or multi-threading to improve system performance and scalability.

By choosing an appropriate data structure and optimizing operations, we can ensure efficient and scalable inventory management system.