**INVENTORY MANAGEMENT SYSTEM**

**Importance of Data Structures and Algorithms**

* **Efficient Storage**: Proper data structures ensure that inventory data is stored in a way that uses memory efficiently.
* **Fast Retrieval**: Efficient algorithms allow for quick access to inventory data, which is critical in a warehouse environment where timely information is crucial.
* **Scalability**: As the inventory grows, efficient data structures and algorithms help maintain performance, preventing slowdowns that could affect warehouse operations.

**Suitable Data Structures**

* **ArrayList**: Provides fast random access but slow insertions and deletions (O(n) time complexity).
* **HashMap**: Offers average O(1) time complexity for insertions, deletions, and lookups, making it a good choice for managing large datasets.
* **TreeMap**: Ensures elements are sorted and provides O(log n) time complexity for basic operations.

**Time Complexity Analysis**

* **Add Operation**: O(1) average time complexity for adding a product to the HashMap.
* **Update Operation**: O(1) average time complexity for updating product attributes in the HashMap.
* **Delete Operation**: O(1) average time complexity for removing a product from the HashMap.

**Optimization Strategies**

* **Efficient Hash Functions**: Use a well-designed hash function to minimize collisions, ensuring operations remain close to O(1).
* **Load Factor Management**: Maintain an appropriate load factor to balance memory usage and performance. Resize the hash table when the load factor exceeds a threshold.
* **Batch Processing**: If updates or deletions are frequently required, consider batch processing to reduce the overhead associated with individual operations.