Why Data Structures and Algorithms Are Essential

Data structures and algorithms are critical in inventory management systems for several reasons:

- **Efficiency**: Proper data structures ensure efficient data storage, retrieval, and manipulation, which is crucial for large inventories.
- **Scalability**: The choice of data structure affects the system's ability to scale with an increasing number of products.
- **Optimization**: Algorithms determine the efficiency of operations like searching, sorting, adding, updating, and deleting items.

Suitable Data Structures

The choice of data structure depends on the operations and requirements of the system:

1. **ArrayList**: Useful for maintaining a list of products where order matters and there are frequent iterations over the list.
2. **HashMap**: Ideal for quick access, addition, and deletion of products using a unique key (e.g., productId). It provides average constant time complexity for these operations.
3. **TreeMap**: Provides sorted order based on keys, with logarithmic time complexity for operations. Suitable if sorted access is needed.

Time Complexity

For this exercise, we'll use a HashMap because it allows efficient storage and retrieval using a unique identifier (productId).

- **Addition (addProduct)**: The average time complexity for adding a product is O(1) because HashMap provides constant-time complexity for put operations. However, in the worst case (due to hash collisions), it could be O(n).
- **Update (updateProduct)**: Similar to addition, updating a product also takes O(1) on average and O(n) in the worst case.
- **Deletion (deleteProduct)**: Deleting a product by key also generally takes O(1) time on average, with a worst-case of O(n).

Optimization

To optimize these operations:

1. **Use a Good Hash Function**: Ensure that the hash function distributes keys uniformly to minimize collisions.
2. **Efficient Handling of Collisions**: Use techniques like separate chaining or open addressing to handle hash collisions effectively.