# Inventory Management System - Analysis

## 4. Analysis

Time Complexity Analysis:

--------------------------

Assuming we use a HashMap (e.g., HashMap<String, Product>) to store the inventory with productId as the key:

1. Add Product:
   - Time Complexity: O(1) on average (due to direct hashing)
   - Justification: Inserting into a HashMap requires a hash computation and placing the item in the corresponding bucket.

2. Update Product:
   - Time Complexity: O(1) on average
   - Justification: Finding a product by key and updating its details is done in constant time in a HashMap.

3. Delete Product:
   - Time Complexity: O(1) on average
   - Justification: Deleting by key in a HashMap is done in constant time after locating the key.

Optimization Discussion:

-------------------------

- Ensure a well-distributed hash function to minimize collisions and maintain O(1) efficiency.
- Use LinkedHashMap if ordering is also needed along with fast access.

# Inventory Management System - Analysis

- For range queries (e.g., find all products within a price range), consider using TreeMap or additional data structures like PriorityQueues.
- For very large inventories, caching frequently accessed products and batch processing updates can improve performance.