**Understand the Problem**

* **Why Data Structures and Algorithms Matter:**

When managing a warehouse inventory, you may deal with **hundreds or thousands of products**. Without an efficient way to **store and retrieve** product information, your system can become slow, confusing, and hard to maintain.

This is where **data structures** and **algorithms** come into play. They help ensure that operations like **adding a new product**, **updating stock**, or **removing discontinued items** are done quickly and efficiently.

**Key Reasons:**

* **Fast Lookups:**  
  You need to check product details or update quantities instantly. A good data structure reduces the time it takes to search for a product.
* **Quick Deletions:**  
  If a product is no longer available, the system should remove it without scanning the entire inventory.
* **Efficient Memory Usage:**  
  Using the right structure avoids wasting memory or duplicating data unnecessarily.
* **Clean and Organized Code:**  
  Proper data structures make your code easier to read, debug, and extend in the future.
* **Suitable Data Structures for This Problem:**

There are a few common data structures you could use, but some are more appropriate than others depending on your needs:

1. **ArrayList**
   * Stores elements in order.
   * Easy to iterate and display.
   * **Drawback:** Searching for a product by ID is slow, with a time complexity of **O(n)**, because it must scan each item one by one.
2. **HashMap** (Best choice for this case)
   * Stores items as **key-value pairs**.
   * The key can be the productId, and the value can be a Product object.
   * Offers **constant time complexity (O(1))** for operations like **add**, **search**, and **delete**.
   * Great for quick lookups when you know the product ID.