# **WEEK 1 DATA STRUCTURES AND ALGORITHMS HANDS ON**

**SUBMITTED BY :-**

NAME – AYUSH SAHOO

SUPERSET ID – 6372243

COURSE – DIGITAL NURTURE 4.0 JAVA FSE

EMAIL – [ayush.sahoo003@gmail.com](mailto:ayush.sahoo003@gmail.com)

COLLEGE – C.V. RAMAN GLOBAL UNIVERSITY

**Exercise 2: E-commerce Platform Search Function**

**Scenario:**

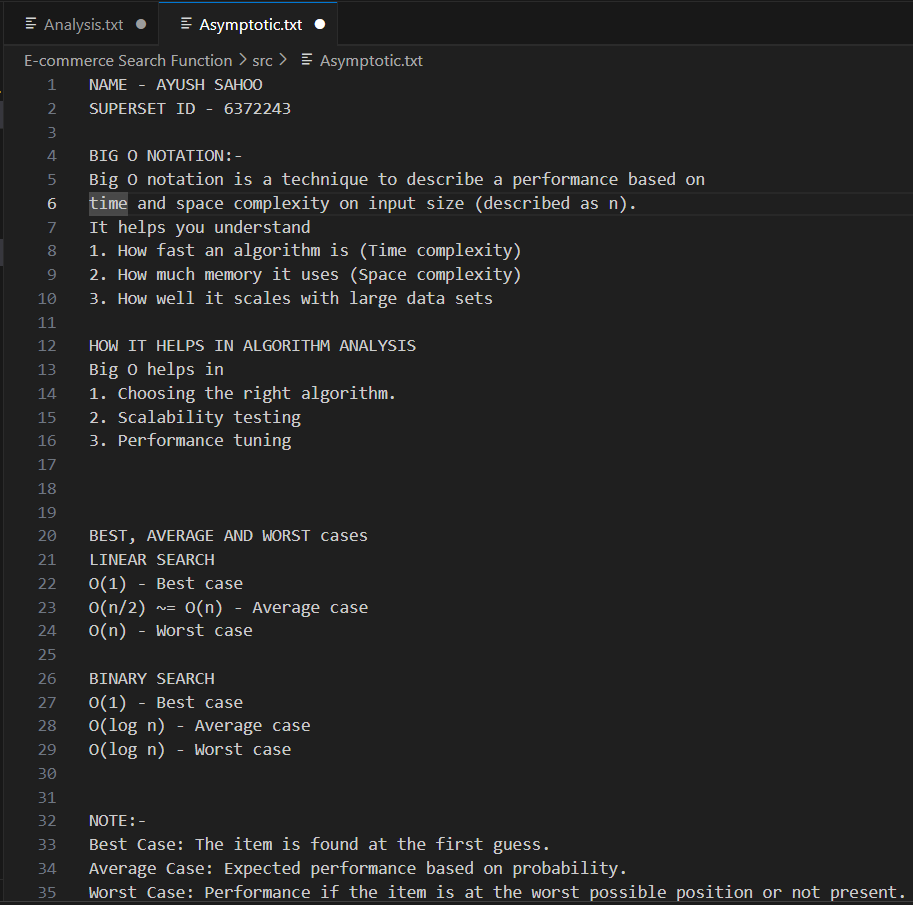
**You are working on the search functionality of an e-commerce platform. The search needs to be optimized for fast performance.**

**Steps:**

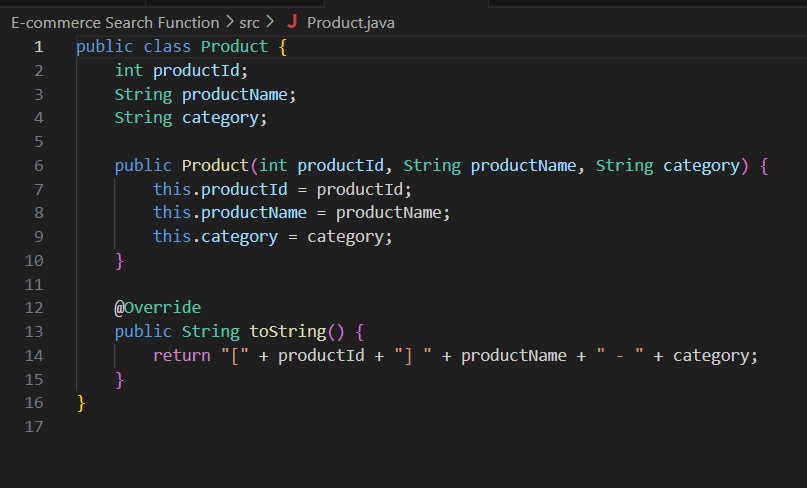
1. **Understand Asymptotic Notation:**
   * **Explain Big O notation and how it helps in analyzing algorithms.**
   * **Describe the best, average, and worst-case scenarios for search operations.**
2. **Setup:**
   * **Create a class Product with attributes for searching, such as productId, productName, and category.**
3. **Implementation:**
   * **Implement linear search and binary search algorithms.**
   * **Store products in an array for linear search and a sorted array for binary search.**
4. **Analysis:**
   * **Compare the time complexity of linear and binary search algorithms.**
   * **Discuss which algorithm is more suitable for your platform and why.**

**SOUTION**

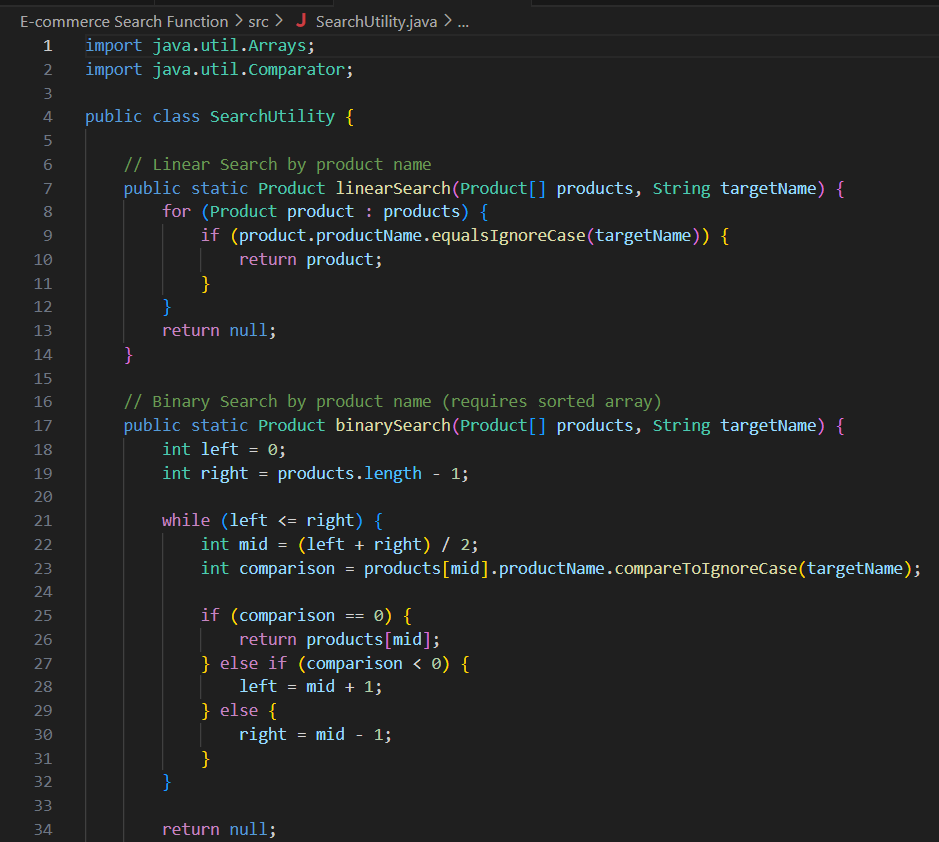
**STEP 1:**

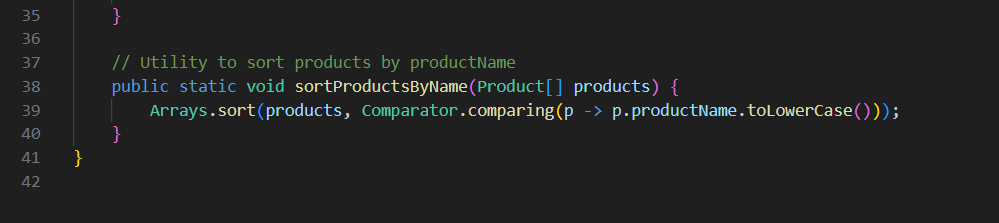


**STEP 2:**

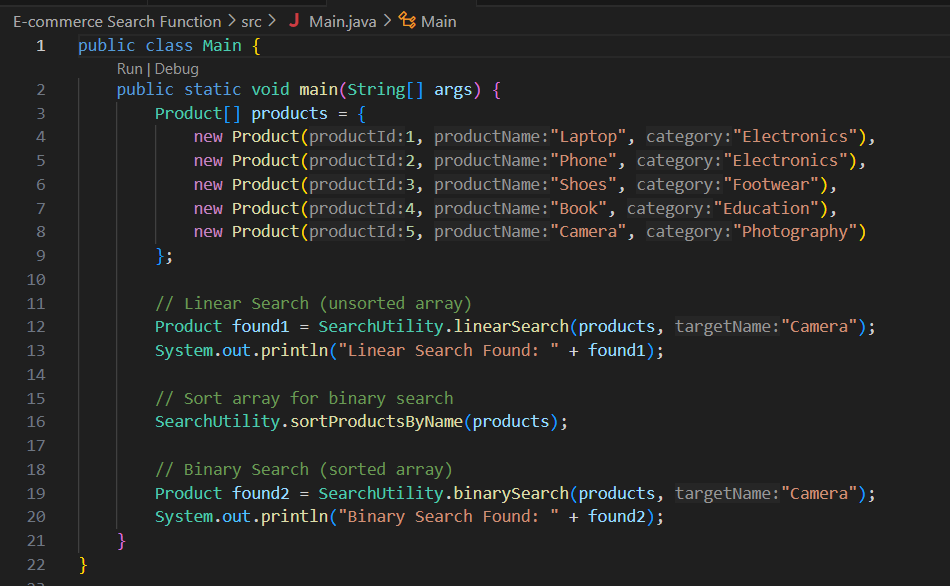


**STEP 3:**

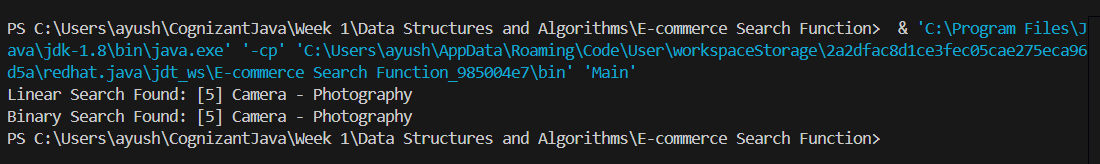
****

****

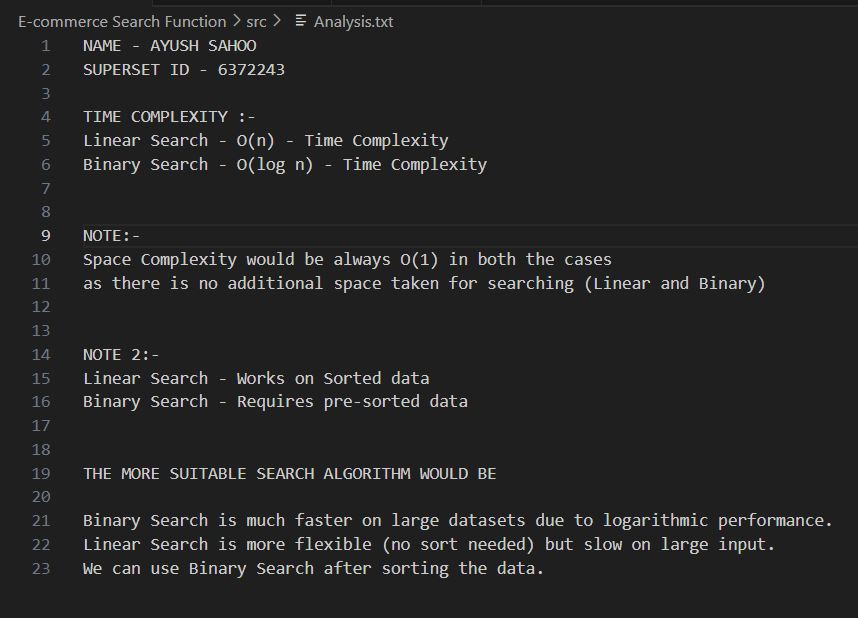
**STEP 4:**

****

**OUTPUT:**



**STEP 5:**

****

**Exercise 7: Financial Forecasting**

**Scenario:**

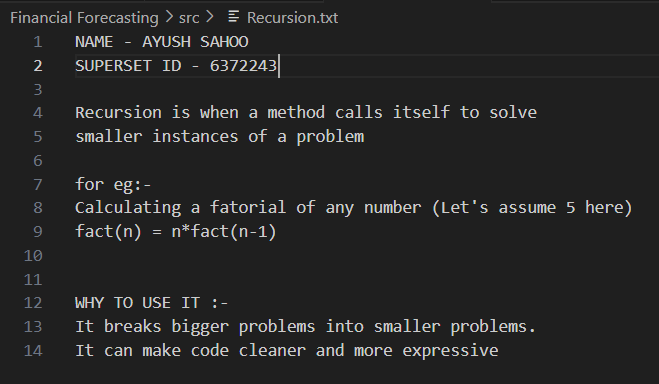
**You are developing a financial forecasting tool that predicts future values based on past data.**

**Steps:**

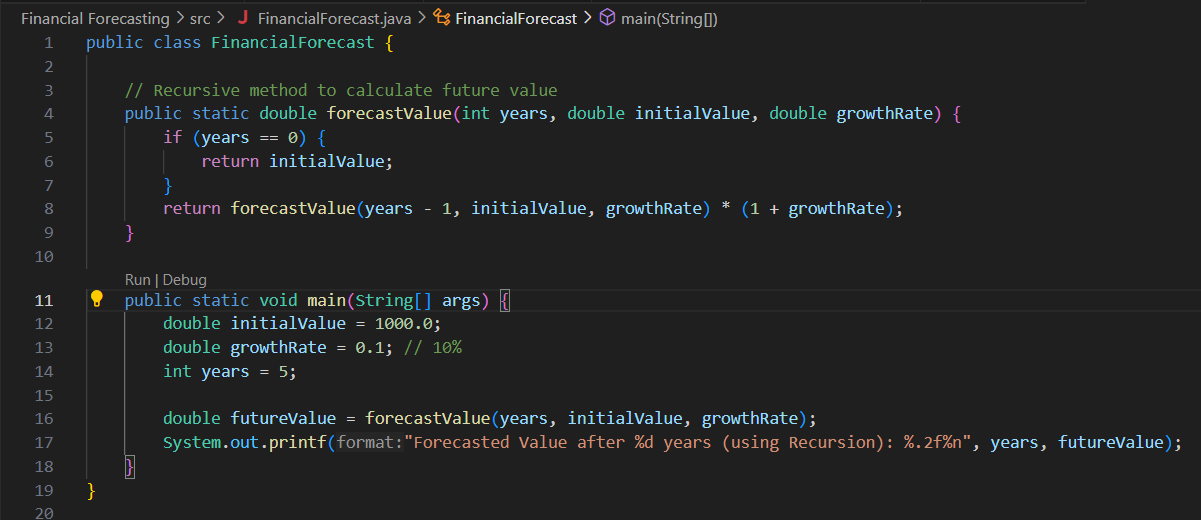
1. **Understand Recursive Algorithms:**
   * **Explain the concept of recursion and how it can simplify certain problems.**
2. **Setup:**
   * **Create a method to calculate the future value using a recursive approach.**
3. **Implementation:**
   * **Implement a recursive algorithm to predict future values based on past growth rates.**
4. **Analysis:**
   * **Discuss the time complexity of your recursive algorithm.**
   * **Explain how to optimize the recursive solution to avoid excessive computation.**

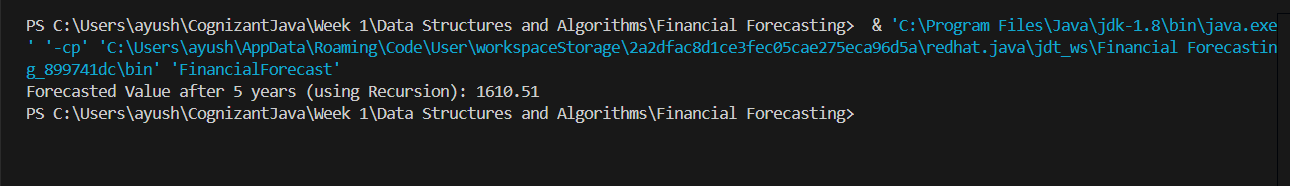
**SOLUTION:**

**STEP 1:**

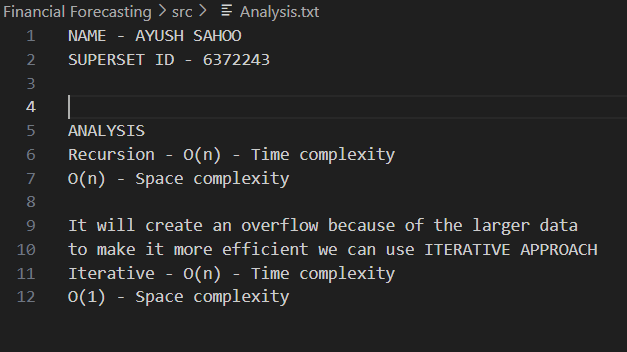


**STEP 2 AND STEP 3:**



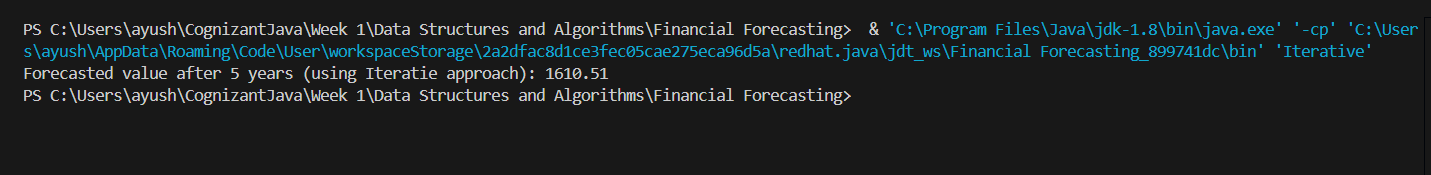
**OUTPUT:**

**STEP 4.1 ANALYSIS:**



**STEP 4.2 BETTER APPROACH (ITERATIVE APPROACH):**



**OUTPUT:**