E-commerce Platform Search Function (C#)

# 1. Overview

This project demonstrates the implementation of search functionality for an e-commerce platform using C#. It includes both Linear Search and Binary Search algorithms for finding products by name.

# 2. Asymptotic Notation

Big O notation is used to describe the performance of an algorithm in terms of time or space.

Time Complexities for Search Algorithms:

* Linear Search: Best - O(1), Average - O(n), Worst - O(n)
* Binary Search: Best - O(1), Average - O(log n), Worst - O(log n)

# 3. Product Class

This class represents each product with its ID, name, and category.

public class Product  
{  
 public int ProductId { get; set; }  
 public string ProductName { get; set; }  
 public string Category { get; set; }  
  
 public Product(int id, string name, string category)  
 {  
 ProductId = id;  
 ProductName = name;  
 Category = category;  
 }  
}

# 4. Search Class

This class contains both Linear and Binary Search methods.

public class Search  
{  
 public static Product LinearSearch(Product[] products, string targetName)  
 {  
 foreach (var product in products)  
 {  
 if (product.ProductName.Equals(targetName, StringComparison.OrdinalIgnoreCase))  
 {  
 return product;  
 }  
 }  
 return null;  
 }  
  
 public static Product BinarySearch(Product[] products, string targetName)  
 {  
 int left = 0, right = products.Length - 1;  
  
 while (left <= right)  
 {  
 int mid = (left + right) / 2;  
 int comparison = string.Compare(products[mid].ProductName, targetName, StringComparison.OrdinalIgnoreCase);  
  
 if (comparison == 0)  
 return products[mid];  
 else if (comparison < 0)  
 left = mid + 1;  
 else  
 right = mid - 1;  
 }  
 return null;  
 }  
}

# 5. Main Program

The main function creates a list of products and demonstrates both search algorithms.

class Program  
{  
 static void Main(string[] args)  
 {  
 Product[] products = new Product[]  
 {  
 new Product(101, "Phone", "Electronics"),  
 new Product(102, "Shirt", "Clothing"),  
 new Product(103, "Laptop", "Electronics"),  
 new Product(104, "Book", "Education")  
 };  
  
 var result1 = Search.LinearSearch(products, "Laptop");  
 Console.WriteLine(result1 != null ? $"Found (Linear): {result1.ProductName}" : "Not Found (Linear)");  
  
 var sortedProducts = products.OrderBy(p => p.ProductName).ToArray();  
 var result2 = Search.BinarySearch(sortedProducts, "Laptop");  
 Console.WriteLine(result2 != null ? $"Found (Binary): {result2.ProductName}" : "Not Found (Binary)");  
 }  
}

