# Exercise 2: E-commerce Platform Search Function (.NET)

## What I Learned

In this task, I built the search functionality of an e-commerce platform using C#. I understood and applied Big O notation to analyze the efficiency of different search algorithms. I implemented both linear and binary search for searching products and compared their performance.

## Concepts Covered

- Big O Notation

- Linear vs Binary Search

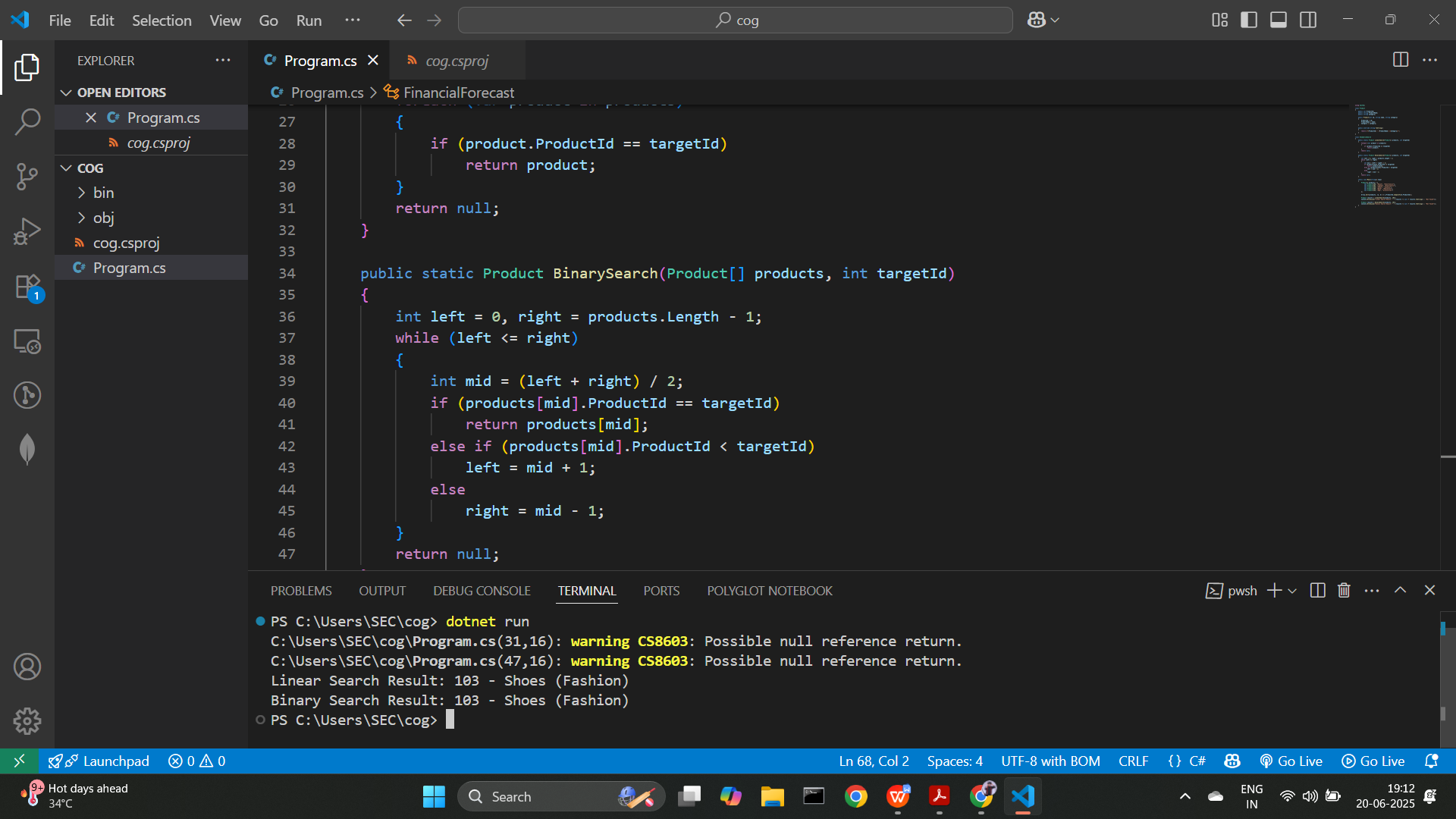
- Time complexity: Best, Average, and Worst Cases

- Sorting before Binary Search

## C# Code

using System;  
  
class Product  
{  
 public int ProductId;  
 public string ProductName;  
 public string Category;  
  
 public Product(int id, string name, string category)  
 {  
 ProductId = id;  
 ProductName = name;  
 Category = category;  
 }  
  
 public override string ToString()  
 {  
 return $"{ProductId} - {ProductName} ({Category})";  
 }  
}  
  
class ECommerceSearch  
{  
 public static Product LinearSearch(Product[] products, int targetId)  
 {  
 foreach (var product in products)  
 {  
 if (product.ProductId == targetId)  
 return product;  
 }  
 return null;  
 }  
  
 public static Product BinarySearch(Product[] products, int targetId)  
 {  
 int left = 0, right = products.Length - 1;  
 while (left <= right)  
 {  
 int mid = (left + right) / 2;  
 if (products[mid].ProductId == targetId)  
 return products[mid];  
 else if (products[mid].ProductId < targetId)  
 left = mid + 1;  
 else  
 right = mid - 1;  
 }  
 return null;  
 }  
  
 static void Main(string[] args)  
 {  
 Product[] products = {  
 new Product(105, "Mouse", "Electronics"),  
 new Product(101, "Laptop", "Electronics"),  
 new Product(103, "Shoes", "Fashion"),  
 new Product(104, "Book", "Education"),  
 new Product(102, "Bag", "Accessories")  
 };  
  
 Array.Sort(products, (a, b) => a.ProductId.CompareTo(b.ProductId));  
  
 Product result1 = LinearSearch(products, 103);  
 Console.WriteLine("Linear Search Result: " + (result1 != null ? result1.ToString() : "Not Found"));  
  
 Product result2 = BinarySearch(products, 103);  
 Console.WriteLine("Binary Search Result: " + (result2 != null ? result2.ToString() : "Not Found"));  
 }  
}

## Output



## Conclusion

Binary search gave faster results than linear search but required the array to be sorted first. This helped me understand how algorithm efficiency impacts application performance, especially in large-scale systems like e-commerce platforms.