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

**Time Complexity :**

Big O notation helps us understand how efficient an algorithm is, especially when the number of items grows. It describes how the time taken by an algorithm increases as the input size increases, without getting into specific hardware or exact timing.

For example:

* **O(1)**: Constant time.
* **O(n)**: Linear time.
* **O(log n)**: Logarithmic time.

**Source Code :**

import java.util.\*;

class Dish {

    int id;

    String name;

    String cuisine;

    Dish(int id, String name, String cuisine) {

        this.id = id;

        this.name = name;

        this.cuisine = cuisine;

    }

    public String toString() {

        return id + " - " + name + " (" + cuisine + ")";

    }

}

public class EcommercePlatformSearchFunction {

    public static void main(String[] args) {

        Dish[] menu = {

            new Dish(101, "Butter Chicken", "Indian"),

            new Dish(102, "Margherita Pizza", "Italian"),

            new Dish(103, "Chicken Tacos", "Mexican"),

            new Dish(104, "Pad Thai", "Thai"),

            new Dish(105, "Cheeseburger", "American"),

            new Dish(106, "Ramen", "Japanese")

        };

        Arrays.sort(menu, Comparator.comparing(d -> d.name.toLowerCase()));

        Scanner scanner = new Scanner(System.in);

        System.out.println("Welcome to FoodFiesta!");

        System.out.print("What dish are you craving today? ");

        String searchQuery = scanner.nextLine();

        boolean found = false;

        int low = 0, high = menu.length - 1;

        while (low <= high) {

            int mid = (low + high) / 2;

            int compare = searchQuery.compareToIgnoreCase(menu[mid].name);

            if (compare == 0) {

                System.out.println("Found using binary search: " + menu[mid]);

                found = true;

                break;

            } else if (compare < 0) {

                high = mid - 1;

            } else {

                low = mid + 1;

            }

        }

        for (Dish dish : menu) {

            if (dish.name.equalsIgnoreCase(searchQuery)) {

                System.out.println("Found using linear search: " + dish);

                found = true;

                break;

            }

        }

        if (!found) {

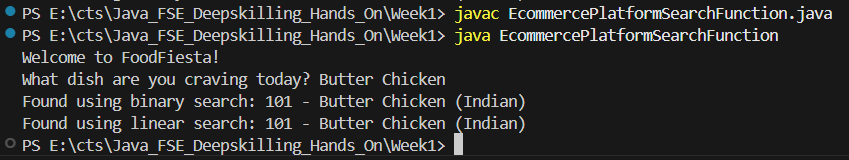
            System.out.println("Sorry, we couldn't find that dish on our menu.");

        }

    }

}

**Output :**

****