**Q1: E-commerce Product Search with Binary Search**: Implement a **binary search** algorithm to search for products in a sorted product catalog. Compare its performance against **linear search**.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>E-commerce Product Search with Binary Search</title>

<style>

body { font-family: Arial, sans-serif; padding: 20px; }

input { padding: 10px; margin: 10px 0; }

button { padding: 10px 15px; cursor: pointer; }

.result { margin-top: 20px; font-weight: bold; }

</style>

</head>

<body>

<h1>Product Search</h1>

<input type="text" id="productName" placeholder="Enter product name...">

<button onclick="searchProduct()">Search</button>

<div class="result" id="result"></div>

<script>

const products = ["Apple", "Banana", "Cherry", "Grapes", "Lemon", "Mango", "Orange", "Peach", "Pineapple", "Strawberry"];

function binarySearch(arr, target) {

let left = 0;

let right = arr.length - 1;

while (left <= right) {

const mid = Math.floor((left + right) / 2);

if (arr[mid] === target) return mid;

if (arr[mid] < target) left = mid + 1;

else right = mid - 1;

}

return -1;

}

function linearSearch(arr, target) {

for (let i = 0; i < arr.length; i++) {

if (arr[i] === target) return i;

}

return -1;

}

function searchProduct() {

const product = document.getElementById("productName").value.trim();

const binaryResult = binarySearch(products, product);

const linearResult = linearSearch(products, product);

const resultElement = document.getElementById("result");

if (binaryResult !== -1) {

resultElement.innerHTML = `Product "${product}" found at index ${binaryResult} using Binary Search.<br>Linear Search also found it at index ${linearResult}.`;

} else {

resultElement.innerHTML = `Product "${product}" not found.`;

}

}

</script>

</body>

</html>