**Q2: Contact List Sorting (Merge Sort)**: Sort a large list of phone contacts using **merge sort** and compare the time complexity with **quick sort** when applied to smaller lists.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Contact List Sorting with Merge Sort</title>

<style>

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

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

pre { margin-top: 20px; font-size: 14px; }

</style>

</head>

<body>

<h1>Contact List Sorting</h1>

<button onclick="sortContacts()">Sort Contacts with Merge Sort</button>

<pre id="sortedContacts"></pre>

<script>

const contacts = ["John", "Jane", "Alex", "Chris", "Max", "Sarah", "Mike", "Zoe", "Anna", "Ella"];

function mergeSort(arr) {

if (arr.length <= 1) return arr;

const mid = Math.floor(arr.length / 2);

const left = mergeSort(arr.slice(0, mid));

const right = mergeSort(arr.slice(mid));

return merge(left, right);

}

function merge(left, right) {

let result = [];

let i = 0, j = 0;

while (i < left.length && j < right.length) {

if (left[i] < right[j]) result.push(left[i++]);

else result.push(right[j++]);

}

return result.concat(left.slice(i), right.slice(j));

}

function sortContacts() {

const start = performance.now();

const sorted = mergeSort(contacts);

const end = performance.now();

document.getElementById("sortedContacts").textContent = `Sorted Contacts: ${sorted.join(", ")}\nExecution time: ${(end - start).toFixed(4)}ms`;

}

</script>

</body>

</html>