

**Shri Vile Parle Kelavani Mandal’s**

**Institute of Technology, Dhule**

**Department of Information Technology**

**Design and Analysis of Algorithm Lab**

**Name :** Yukta Durgesh Pawar

**Class :** SY ( IT )

**PRN :** 2254491246051

**Roll No :**  52

**Problem :** Write an algorithm for merge sort. Apply merge sort on following array.

A = 5 1 2 6 3 7 9 4

**Solution :**

**Merge Sort :**

* Merge sort is yet another sorting algorithm that falls under the category of **Divide and Conquer** technique.
* In this technique, we segment a problem into two halves and solve them individually.
* After finding the solution of each half, we merge them back to represent the solution of the main problem.
* It is one of the best sorting techniques that successfully build a recursive algorithm.
* Merge sort was developed by Jon Von Neumann in 1945 and it has O(n log n) performance in both worst and average case.
* The main task in merge sort is merging two sorted lists into a single sorted list.

**Algorithm Of Merge Sort :**

This algorithm is based on the divide and conquer technique. The list is recursively divided till we get single element lists which are obviously sorted and then the lists are merged repeatedly to get a single sorted list.

**Step - 1 :** Divide the list into two sublists of almost equal size.

**Step - 2 :** Sort the left sublist recursively using merge sort.

**Step - 3 :** Sort the right sublist recursively using merge sort.

**Step - 4 :** Merge the two sorted sublists.

Terminating condition for recursion is when the sublist formed contains only one element. If the list contains odd number of elements then we assume that the left half is bigger.

Let’s understand the merge sort by following example in which the merge sort algorithm recursively divides the array into halves until the base condition is met, where we are left with only 1 element in the array. And then, the merge function picks up the sorted sub-arrays and merge them back to sort the entire array.

**Example :**

Consider the following example of an unsorted array, which we are going to sort with the help of the Merge Sort algorithm.

A = 5 1 2 6 3 7 9 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 1 | 2 | 6 | 3 | 7 | 9 | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 1 | 2 | 6 |

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | 7 | 9 | 4 |

|  |  |
| --- | --- |
| 5 | 1 |

|  |  |
| --- | --- |
| 2 | 6 |

|  |  |
| --- | --- |
| 3 | 7 |

|  |  |
| --- | --- |
| 9 | 4 |

|  |
| --- |
| 5 |

|  |
| --- |
| 1 |

|  |
| --- |
| 2 |

|  |
| --- |
| 6 |

|  |
| --- |
| 3 |

|  |
| --- |
| 7 |

|  |
| --- |
| 9 |

|  |
| --- |
| 4 |

Split

Merge

|  |
| --- |
| 5 |

|  |
| --- |
| 1 |

|  |
| --- |
| 2 |

|  |
| --- |
| 6 |

|  |
| --- |
| 3 |

|  |
| --- |
| 7 |

|  |
| --- |
| 9 |

|  |
| --- |
| 4 |

|  |  |
| --- | --- |
| 1 | 5 |

|  |  |
| --- | --- |
| 2 | 6 |

|  |  |
| --- | --- |
| 3 | 7 |

|  |  |
| --- | --- |
| 4 | 9 |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 5 | 6 |

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | 4 | 7 | 9 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 |

**Step 1 :**

The merge sort algorithm iteratively divides an array into equal halves until we achieve an atomic value.

**Step 2 :**

After dividing an array into two subarrays, we will notice that it did not hamper the order of elements as they were in the original array. After now, we will further divide these two arrays into other halves.

**Step 3 :**

Again, we will divide these arrays until we achieve an atomic value, i.e., a value that cannot be further divided

**Step 4 :**

Next, we will merge them back in the same way as they were broken down.

**Step 5 :**

For each list, we will first compare the element and then combine them to form a new sorted list.

**Step 6 :**

In the next iteration, we will compare the lists of two data values and merge them back into a list of found data values, all placed in a sorted manner.