```
package datastructure.Sort;

public class MergeSort {
    int[] array;
    int[] tempArray;

    void arrangementForSort(int[] arr) {
        this.array = arr;
        this.tempArray = new int[arr.length];
        doMergeSort(0, arr.length - 1);
    }

    void doMergeSort(int low, int high) {
        if (low < high) {
            int middle = low + (high - low) / 2;
            doMergeSort(low, middle);
            doMergeSort(middle + 1, high);
            MergePart(low, middle, high);
            System.out.println();
        }
    }
    //  0   1   2   3
    //  4,  1,  7,  3,

    void MergePart(int low, int middle, int high) {
        for (int i = low; i <= high; i++)
            tempArray[i] = array[i];
        int i = low;
```

```
        int j = middle + 1;
        int k = low;
        while (i <= middle && j <= high) {
            if (tempArray[i] <= tempArray[j]) {
                array[k] = tempArray[i];
                i++;
            } else {
                array[k] = tempArray[j];
                j++;
            }
            k++;
        }
        while (i <= middle) {
            array[k] = tempArray[i];
            k++;
            i++;
        }
    }

    public static void main(String[] args) {
        int[] arr = { 4, 1, 7, 3 };
        new MergeSort().arrangementForSort(arr);
        System.out.println("\nAfter sorting");
        for (int i = 0; i < arr.length; i++)
            System.out.print(arr[i] + "\t");
        System.out.println();
    }
```

```
}
```