

MergeSort.java

```

1 package in.neuron.main;
2
3 public class MergeSort {
4     void merge(int arr[], int p, int q, int r) {
5
6
7         int n1 = q - p + 1;
8         int n2 = r - q;
9
10        int L[] = new int[n1];
11        int M[] = new int[n2];
12
13        for (int i = 0; i < n1; i++)
14            L[i] = arr[p + i];
15        for (int j = 0; j < n2; j++)
16            M[j] = arr[q + 1 + j];
17
18
19        int i, j, k;
20        i = 0;
21        j = 0;
22        k = p;
23        while (i < n1 && j < n2) {
24            if (L[i] <= M[j]) {
25                arr[k] = L[i];
26                i++;
27            } else {
28                arr[k] = M[j];
29                j++;
30            }
31            k++;
32        }
33
34        while (i < n1) {
35            arr[k] = L[i];
36            i++;
37            k++;
38        }
39
40        while (j < n2) {
41            arr[k] = M[j];
42            j++;
43            k++;
44        }
45    }
46
47    void mergeSort(int arr[], int l, int r) {
48        if (l < r) {
49
50            int m = (l + r) / 2;
51
52            mergeSort(arr, l, m);
53            mergeSort(arr, m + 1, r);
54            merge(arr, l, m, r);
55        }
56    }
57    static void printArray(int arr[]) {
58        int n = arr.length;
59        for (int i = 0; i < n; ++i)

```

MergeSort.java

```
60         System.out.print(arr[i] + " ");
61     System.out.println();
62 }
63
64 public static void main(String args[]) {
65     int arr[] = {8,6,7,9,2,4,3,1};
66
67     MergeSort ob = new MergeSort();
68     ob.mergeSort(arr, 0, arr.length - 1);
69
70     System.out.println("Sorted array:");
71     printArray(arr);
72
73
74 }
75
76 }
77
```