

Iske baad humme while loop ka use kiya hai jisme hum sorted array ke saare elements ke saath compare karte hain. Jab tak sorted array ke kisi element se "key" bada element milta hai, hum us element ko ek position aage move karte hain.

Jab humme "key" ko correct position par insert kar diya, toh humne ek aur loop ka use kiya hai jiske madhyam se hum sorted array ko print karte hain.

Is tarah se, ye program insertion sort algorithm ka ek simple implementation hai.

ky aap isi tarah Merge sort bhi samjha sakte hai

< 2 / 2

Haan, main aapko merge sort algorithm ke java code ke madhyam se samjha sakta hun. Chaliye, hum merge sort algorithm ko samajh lete hain.

Merge sort algorithm ek divide and conquer algorithm hai, jisme hum array ko recursively half karke sort karte hain. Is algorithm ke pratham kadam mein hum array ko do parts mein divide karte hain, aur fir har part ko phir se divide karte hain, jab tak har part mein sirf ek element na bach jaye. Fir, hum har part ko sort karke dobara merge karte hain, jisse pura array sort ho jata hai.

Yahan, hum ek simple java program likhenge jo merge sort algorithm ka use karta hai. Ye program ek integer array ko lekar usko sort karega.

```
1 public class MergeSortExample {
2     public static void main(String[] args) {
3         int[] array = {5, 2, 9, 1, 3, 0};
4         int n = array.length;
5
6         // Sort the array
7         mergeSort(array, 0, n - 1);
8
9         // Print the sorted array
10        for (int i = 0; i < n; i++) {
11            System.out.print(array[i] + " ");
12        }
13    }
14
15    public static void mergeSort(int[] array, int left, int right) {
16        if (left < right) {
17            // Find the middle point
18            int middle = (left + right) / 2;
19
20            // Sort first and second halves
21            mergeSort(array, left, middle);
22            mergeSort(array, middle + 1, right);
23
24            // Merge the sorted halves
25            merge(array, left, middle, right);
26        }
27    }
28
29    public static void merge(int[] array, int left, int middle, int right) {
30        // Find sizes of two subarrays to be merged
31        int n1 = middle - left + 1;
32        int n2 = right - middle;
33
34        // Create temporary arrays
35        int[] leftArray = new int[n1];
36        int[] rightArray = new int[n2];
37
38        // Copy data to temporary arrays
39        for (int i = 0; i < n1; i++) {
40            leftArray[i] = array[left + i];
41        }
42        for (int j = 0; j < n2; j++) {
43            rightArray[j] = array[middle + 1 + j];
44        }
45
46        // Merge the temporary arrays
47        int i = 0, j = 0;
48        int k = left;
49        while (i < n1 && j < n2) {
50            if (leftArray[i] < rightArray[j]) {
51                array[k] = leftArray[i];
52                i++;
53            } else {
54                array[k] = rightArray[j];
55                j++;
56            }
57            k++;
58        }
59
60        // Copy remaining elements of leftArray[] if any
61        while (i < n1) {
62            array[k] = leftArray[i];
63            i++;
64            k++;
65        }
66
67        // Copy remaining elements of rightArray[] if any
68        while (j < n2) {
69            array[k] = rightArray[j];
70            j++;
71            k++;
72        }
73    }
74 }
```

Is program mein, humne ek Integer array "array" ka use kiya hai jisme humare elements hain. Humne "n" ko array ke length ko use kiya hai.