AlgoExpert Quad Layout Java 12px Sublime Monokai 00:00:00

Run Code

Solution 1 Solution 2

Scratchpad

Prompt

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
 3 class Program {
     // Best: O(nlog(n)) time | O(n) space
      // Average: O(nlog(n)) time | O(n) space
      // Worst: O(nlog(n)) time | O(n) space
      public static int[] mergeSort(int[] array) {
        if (array.length <= 1) {</pre>
          return array;
10
11
        int[] auxiliaryArray = array.clone();
        mergeSort(array, 0, array.length - 1, auxiliaryArray);
12
13
        return array;
14
15
16
      public static void mergeSort(int[] mainArray, int startIdx, int endIdx, int[] auxiliaryArray) {
        if (startIdx == endIdx) {
17
18
          return;
19
        int middleIdx = (startIdx + endIdx) / 2;
20
21
        \verb|mergeSort(auxiliaryArray, startIdx, middleIdx, mainArray);|\\
22
        {\tt mergeSort(auxiliaryArray, middleIdx + 1, endIdx, mainArray);}
23
        \verb|doMerge(mainArray, startIdx, middleIdx, endIdx, auxiliaryArray);|\\
24
25
26
      public static void doMerge(
27
         int[] mainArray, int startIdx, int middleIdx, int endIdx, int[] auxiliaryArray) {
28
        int k = startIdx;
29
        int i = startIdx;
30
        int j = middleIdx + 1;
31
        while (i <= middleIdx && j <= endIdx) {</pre>
32
          if (auxiliaryArray[i] <= auxiliaryArray[j]) {</pre>
            mainArray[k++] = auxiliaryArray[i++];
33
34
          } else {
35
            mainArray[k++] = auxiliaryArray[j++];
36
37
        while (i <= middleIdx) {</pre>
38
          mainArray[k++] = auxiliaryArray[i++];
39
40
41
42
        while (j <= endIdx) {</pre>
          mainArray[k++] = auxiliaryArray[j++];
43
45 }
```

Our Solution(s) Video Explanation