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

Our Solution(s) Run Code Scratchpad Video Explanation

Solution 1 Solution 2

Prompt

40

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
    import java.util.Arrays;
 5 class Program {
     // Best: O(nlog(n)) time | O(nlog(n)) space
      // Average: O(nlog(n)) time | O(nlog(n)) space // Worst: O(nlog(n)) time | O(nlog(n)) space
      public static int[] mergeSort(int[] array) {
         if (array.length <= 1) {</pre>
10
11
          return array;
12
13
         int middleIdx = array.length / 2;
         int[] leftHalf = Arrays.copyOfRange(array, 0, middleIdx);
14
15
         int[] rightHalf = Arrays.copyOfRange(array, middleIdx, array.length);
16
         return mergeSortedArrays(mergeSort(leftHalf), mergeSort(rightHalf));
17
18
      public static int[] mergeSortedArrays(int[] leftHalf, int[] rightHalf) {
19
20
21
         int[] sortedArray = new int[leftHalf.length + rightHalf.length];
         int k = 0;
22
         int i = 0;
23
         int j = 0;
24
         while (i < leftHalf.length && j < rightHalf.length) {</pre>
25
          if (leftHalf[i] <= rightHalf[j]) {</pre>
26
            sortedArray[k++] = leftHalf[i++];
27
28
29
           } else {
             sortedArray[k++] = rightHalf[j++];
30
31
         while (i < leftHalf.length) {</pre>
32
          sortedArray[k++] = leftHalf[i++];
33
34
35
         while (j < rightHalf.length) {</pre>
           sortedArray[k++] = rightHalf[j++];
36
         return sortedArray;
38
39 }
```