Our Solution(s) Run Code

Solution 1 Solution 2

```
1  // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3
4  public class Program {
5    // Best: O(nlog(n)) time | O(n) space
6    // Average: O(nlog(n)) time | O(n) space
7    // Worst: O(nlog(n)) time | O(n) space
8    public static int[] MergeSort(int[] array) {
9      if (array.Length <= 1) {
10       return array;
11     }
12     int[] auxiliaryArray = (int[]) array.Clone();
13     MergeSort(array, 0, array.Length - 1, auxiliaryArray);
14     return array;
15   }
16
17   public static void MergeSort(int[] mainArray, int startIdx, int endI
18     int[] auxiliaryArray) {
19     if (startIdx == endIdx) {
20       return;
21     }
22     int middleIdx = (startIdx + endIdx) / 2;
23     MergeSort(auxiliaryArray, startIdx, middleIdx, mainArray);
24     MergeSort(auxiliaryArray, middleIdx + 1, endIdx, mainArray);
25     doMerge(mainArray, startIdx, middleIdx, endIdx, auxiliaryArray);
26   }
27
28   public static void doMerge(int[] mainArray, int startIdx, int middle
29     int[] auxiliaryArray) {
30     int k = startIdx;
31     int i = startIdx;
32     int j = middleIdx + 1;
33     while (i <= middleIdx && j <= endIdx) {
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
1  public class Program {
2    public static int[] MergeSort(int[] array) {
3      // Write your code here.
4      return null;
5    }
6  }
7
```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.