Solution 1 Solution 2

Run Code

Our Solution(s)

Run Code

Your Solutions

Solution 1 Solution 2 Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
   package main
 5 // Best: O(nlog(n)) time | O(nlog(n)) space
 6 // Average: O(nlog(n)) time | O(nlog(n)) space
   // Worst: O(n\log(n)) time | O(n\log(n)) space
   func MergeSort(array []int) []int {
     if len(array) <= 1 {</pre>
10
       return array
11
      middleIndex := len(array) / 2
12
13
      leftHalf := MergeSort(array[:middleIndex])
14
      rightHalf := MergeSort(array[middleIndex:])
15
      return mergeSortedArrays(leftHalf, rightHalf)
16 }
17
18 func mergeSortedArrays(leftHalf, rightHalf []int) []int {
19
      sortedArray := make([]int, len(leftHalf)+len(rightHalf))
20
      k, i, j := 0, 0, 0
21
       \begin{tabular}{ll} for i < len(leftHalf) && j < len(rightHalf) \end{tabular} 
22
        if leftHalf[i] <= rightHalf[j] {</pre>
23
          sortedArray[k] = leftHalf[i]
24
        } else {
26
          sortedArray[k] = rightHalf[j]
27
28
29
30
31
      for i < len(leftHalf) {</pre>
32
        sortedArray[k] = leftHalf[i]
33
```

```
package main

func MergeSort(array []int) []int {
    // Write your code here.
    return nil
}
```

 Our Tests
 Custom Output
 Submit Code

Run or submit code when you're ready.