

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
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 func MergeSort(array []int) []int {
9     if len(array) <= 1 {
10         return array
11     }
12     auxiliaryArray := make([]int, len(array))
13     copy(auxiliaryArray, array)
14     mergeSortHelper(array, 0, len(array)-1, auxiliaryArray)
15     return array
16 }
17
18 func mergeSortHelper(mainArray []int, startIdx, endIdx int, auxiliaryArray []int) {
19     if startIdx == endIdx {
20         return
21     }
22     middleIdx := (startIdx + endIdx) / 2
23     mergeSortHelper(auxiliaryArray, startIdx, middleIdx, mainArray)
24     mergeSortHelper(auxiliaryArray, middleIdx+1, endIdx, mainArray)
25     doMerge(mainArray, startIdx, middleIdx, endIdx, auxiliaryArray)
26 }
27
28 func doMerge(mainArray []int, startIdx, middleIdx, endIdx int, auxiliaryArray []int) {
29     k := startIdx
30     i := startIdx
31     j := middleIdx + 1
32     for i <= middleIdx && j <= endIdx {
33         if auxiliaryArray[i] <= auxiliaryArray[j] {
34             mainArray[k] = auxiliaryArray[i]
35             i++
36         } else {
37             mainArray[k] = auxiliaryArray[j]
38             j++
39         }
40         k++
41     }
42     for i <= middleIdx {
43         mainArray[k] = auxiliaryArray[i]
44         i++
45         k++
46     }
47     for j <= endIdx {
48         mainArray[k] = auxiliaryArray[j]
49         j++
50         k++
51     }
52 }
53
```

