```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

// 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
func MergeSort(array []int) []int {
  if len(array) <= 1 {
    return array
  }
  middleIndex := len(array) / 2
  leftHalf := MergeSort(array[:middleIndex])
  rightHalf := MergeSort(array[middleIndex:])
  return mergeSortedArrays(leftHalf, rightHalf)
}

func mergeSortedArrays(leftHalf, rightHalf []int) []int {
  sortedArray := make([]int, len(leftHalf)+len(rightHalf))
  k, i, j := 0, 0, 0
  for i < len(leftHalf) && j < len(rightHalf) {
    if leftHalf[i] <= rightHalf[j] {
      sortedArray[k] = leftHalf[i]
      i++
    } else {
      sortedArray[k] = rightHalf[j]
      j++
    }
    k++
  }
  for i < len(leftHalf) {
    sortedArray[k] = leftHalf[i]
    i++
    k++
  }
  for j < len(rightHalf) {
    sortedArray[k] = rightHalf[j]
    j++
    k++
  }
  return sortedArray
}

```