```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# Best: O(nlog(n)) time | O(n) space
# Average: O(nlog(n)) time | O(n) space
# Worst: O(nlog(n)) time | O(n) space
def mergeSort(array):
    if len(array) <= 1:
        return array
    auxiliaryArray = array[:]
    mergeSortHelper(array, 0, len(array) - 1, auxiliaryArray)
    return array


def mergeSortHelper(mainArray, startIdx, endIdx, auxiliaryArray):
    if startIdx == endIdx:
        return
    middleIdx = (startIdx + endIdx) // 2
    mergeSortHelper(auxiliaryArray, startIdx, middleIdx, mainArray)
    mergeSortHelper(auxiliaryArray, middleIdx + 1, endIdx, mainArray)
    doMerge(mainArray, startIdx, middleIdx, endIdx, auxiliaryArray)


def doMerge(mainArray, startIdx, middleIdx, endIdx, auxiliaryArray):
    k = startIdx
    i = startIdx
    j = middleIdx + 1
    while i <= middleIdx and j <= endIdx:
        if auxiliaryArray[i] <= auxiliaryArray[j]:
            mainArray[k] = auxiliaryArray[i]
            i += 1
        else:
            mainArray[k] = auxiliaryArray[j]
            j += 1
        k += 1
    while i <= middleIdx:
        mainArray[k] = auxiliaryArray[i]
        i += 1
        k += 1
    while j <= endIdx:
        mainArray[k] = auxiliaryArray[j]
        j += 1
        k += 1

```