

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // Best: O(nlog(n)) time | O(n) space
4 // Average: O(nlog(n)) time | O(n) space
5 // Worst: O(nlog(n)) time | O(n) space
6 function mergeSort(array) {
7   if (array.length <= 1) return array;
8   const auxiliaryArray = array.slice();
9   mergeSortHelper(array, 0, array.length - 1, auxiliaryArray);
10  return array;
11 }
12
13 function mergeSortHelper(mainArray, startIdx, endIdx, auxiliaryArray) {
14   if (startIdx === endIdx) return;
15   const middleIdx = Math.floor((startIdx + endIdx) / 2);
16   mergeSortHelper(auxiliaryArray, startIdx, middleIdx, mainArray);
17   mergeSortHelper(auxiliaryArray, middleIdx + 1, endIdx, mainArray);
18   doMerge(mainArray, startIdx, middleIdx, endIdx, auxiliaryArray);
19 }
20
21 function doMerge(mainArray, startIdx, middleIdx, endIdx, auxiliaryArray) {
22   let k = startIdx;
23   let i = startIdx;
24   let j = middleIdx + 1;
25   while (i <= middleIdx && j <= endIdx) {
26     if (auxiliaryArray[i] <= auxiliaryArray[j]) {
27       mainArray[k++] = auxiliaryArray[i++];
28     } else {
29       mainArray[k++] = auxiliaryArray[j++];
30     }
31   }
32   while (i <= middleIdx) {
33     mainArray[k++] = auxiliaryArray[i++];
34   }
35   while (j <= endIdx) {
36     mainArray[k++] = auxiliaryArray[j++];
37   }
38 }
39
40 exports.mergeSort = mergeSort;
41
```

