

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // Best: O(nlog(n)) time | O(nlog(n)) space
4 // Average: O(nlog(n)) time | O(nlog(n)) space
5 // Worst: O(nlog(n)) time | O(nlog(n)) space
6 function mergeSort(array) {
7   if (array.length <= 1) return array;
8   const middleIdx = Math.floor(array.length / 2);
9   const leftHalf = array.slice(0, middleIdx);
10  const rightHalf = array.slice(middleIdx);
11  return mergeSortedArrays(mergeSort(leftHalf), mergeSort(rightHalf))
12 }
13
14 function mergeSortedArrays(leftHalf, rightHalf) {
15   const sortedArray = new Array(leftHalf.length + rightHalf.length);
16   let k = 0;
17   let i = 0;
18   let j = 0;
19   while (i < leftHalf.length && j < rightHalf.length) {
20     if (leftHalf[i] <= rightHalf[j]) {
21       sortedArray[k++] = leftHalf[i++];
22     } else {
23       sortedArray[k++] = rightHalf[j++];
24     }
25   }
26   while (i < leftHalf.length) {
27     sortedArray[k++] = leftHalf[i++];
28   }
29   while (j < rightHalf.length) {
30     sortedArray[k++] = rightHalf[j++];
31   }
32   return sortedArray;
33 }
```

Solution 1Solution 2Solution 3

```
1 function mergeSort(array) {
2   // Write your code here.
3 }
4
5 // Do not edit the line below.
6 exports.mergeSort = mergeSort;
7
```

Our Tests

Custom Output

Submit Code

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // Best: O(nlog(n)) time | O(nlog(n)) space
4 // Average: O(nlog(n)) time | O(nlog(n)) space
5 // Worst: O(nlog(n)) time | O(nlog(n)) space
6 function mergeSort(array) {
7   if (array.length <= 1) return array;
8   const middleIdx = Math.floor(array.length / 2);
9   const leftHalf = array.slice(0, middleIdx);
10  const rightHalf = array.slice(middleIdx);
11  return mergeSortedArrays(mergeSort(leftHalf), mergeSort(rightHalf))
12 }
13
14 function mergeSortedArrays(leftHalf, rightHalf) {
15   const sortedArray = new Array(leftHalf.length + rightHalf.length);
16   let k = 0;
17   let i = 0;
18   let j = 0;
19   while (i < leftHalf.length && j < rightHalf.length) {
20     if (leftHalf[i] <= rightHalf[j]) {
21       sortedArray[k++] = leftHalf[i++];
22     } else {
23       sortedArray[k++] = rightHalf[j++];
24     }
25   }
26   while (i < leftHalf.length) {
27     sortedArray[k++] = leftHalf[i++];
28   }
29   while (j < rightHalf.length) {
30     sortedArray[k++] = rightHalf[j++];
31   }
32   return sortedArray;
33 }
```

```
10 # Test Case 40 - Success (1/1)
11 # Test Case 40 - Success (1/1)
12 # Test Case 40 - Success (1/1)
13 # Test Case 40 - Success (1/1)
14 # Test Case 40 - Success (1/1)
15 # Test Case 40 - Success (1/1)
16 # Test Case 40 - Success (1/1)
17 # Test Case 40 - Success (1/1)
18 # Test Case 40 - Success (1/1)
19 # Test Case 40 - Success (1/1)
20 # Test Case 40 - Success (1/1)
```

Run or submit code when you're ready.