

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1	Solution 2	Solution 3
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 class Program { 4 // Best: O(nlog(n)) time O(nlog(n)) space 5 // Average: O(nlog(n)) time O(nlog(n)) space 6 // Worst: O(nlog(n)) time O(nlog(n)) space 7 func mergeSort(_ array: inout [Int]) -> [Int] { 8 if array.count <= 1 { 9 return array 10 } 11 12 let middleIndex = Int(Double(array.count / 2).rounded(.down)) 13 var leftHalf = Array(array[0 ..< middleIndex]) 14 var rightHalf = Array(array[middleIndex ..< array.count]) 15 16 return mergeSortedArrays(mergeSort(&leftHalf), mergeSort(&rightHalf)) 17 } 18 19 func mergeSortedArrays(_ leftHalf: [Int], _ rightHalf: [Int]) -> [Int] { 20 var sortedArray = Array(repeating: 0, count: leftHalf.count + rightHalf.count) 21 22 var k = 0, i = 0, j = 0 23 24 while i < leftHalf.count, j < rightHalf.count { 25 if leftHalf[i] <= rightHalf[j] { 26 sortedArray[k] = leftHalf[i] 27 i += 1 28 } else { 29 sortedArray[k] = rightHalf[j] 30 j += 1 31 } 32 33 k += 1 34 } 35 36 while i < leftHalf.count { 37 sortedArray[k] = leftHalf[i] 38 i += 1 39 k += 1 40 } 41 42 while j < rightHalf.count { 43 sortedArray[k] = rightHalf[j] 44 j += 1 45 k += 1 46 } 47 48 return sortedArray 49 } 50 }</pre>		<pre>1 class Program { 2 func mergeSort(_ array: inout [Int]) -> [Int] { 3 // Write your code here. 4 return [] 5 } 6 } 7</pre>

Our Tests

Custom Output

Submit Code

```
1 class Program {
2     func mergeSort(_ array: inout [Int]) -> [Int] {
3         // Write your code here.
4         return []
5     }
6 }
7
```

```

10 #testcase/Test Case 475 (2) #input = test 20
11 # # #
12 # # #
13 # # #
14 #testcase/Test Case 475 (2) #input = test 20
15 # # #
16 # # #
17 # # #
18 #testcase/Test Case 475 (2) #input = test 20
19 # # #
20 # # #
21 # # #
22 #testcase/Test Case 475 (2) #input = test 20
23 # # #
24 # # #
25 # # #

```

Run or submit code when you're ready.