AlgoExpert Quad Layout Swift 12px Sublime Monokai 00:00:00

 Prompt
 Scratchpad
 Our Solution(s)
 Video Explanation

Run Code

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
    class Program {
       // Best: O(nlog(n) time | O(n) space
       // Average: O(nlog(n) time | O(n) space
       // Worst: O(nlog(n) time | O(n) space
        func mergeSort(_ array: inout [Int]) -> [Int] {
          if array.count <= 1 {</pre>
               return array
10
11
12
           var auxiliaryArray = array
13
           mergeSortHelper(0, array.count - 1, &array, &auxiliaryArray)
14
15
           return array
16
17
18
        19
           if startIndex == endIndex {
20
               return
21
22
23
           let middleIndex = Int(Double((startIndex + endIndex) / 2).rounded(.down))
24
25
           mergeSortHelper(startIndex, middleIndex, &secondArray, &firstArray)
26
           mergeSortHelper(middleIndex + 1, endIndex, &secondArray, &firstArray)
27
           \verb|doMerge(startIndex, middleIndex, endIndex, \&firstArray, \&secondArray)|\\
28
29
30
        func doMerge(_ startIndex: Int, _ middleIndex: Int, _ endIndex: Int, _ firstArray: inout [Int], _ secondArray: inout [Int]) {
31
           var k = startIndex, i = startIndex, j = middleIndex + 1
32
33
           if secondArray[i] <= secondArray[j] {</pre>
34
35
                  firstArray[k] = secondArray[i]
36
                  i += 1
              } else {
37
38
                  firstArray[k] = secondArray[j]
39
                  j += 1
40
41
42
               k += 1
43
44
45
           while i <= middleIndex {</pre>
46
              firstArray[k] = secondArray[i]
47
               i += 1
48
               k += 1
49
50
51
           while j <= endIndex {</pre>
52
               firstArray[k] = secondArray[j]
53
              j += 1
54
               k += 1
55
57 }
```

Solution 1

Solution 2