

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 // Best: O(nlog(n)) time | O(n) space
6 // Average: O(nlog(n)) time | O(n) space
7 // Worst: O(nlog(n)) time | O(n) space
8 func MergeSort(array []int) []int {
9     if len(array) <= 1 {
10         return array
11     }
12     auxiliaryArray := make([]int, len(array))
13     copy(auxiliaryArray, array)
14     mergeSortHelper(array, 0, len(array)-1, auxiliaryArray)
15     return array
16 }
17
18 func mergeSortHelper(mainArray []int, startIdx, endIdx int, auxiliaryA
19     if startIdx == endIdx {
20         return
21     }
22     middleIdx := (startIdx + endIdx) / 2
23     mergeSortHelper(auxiliaryArray, startIdx, middleIdx, mainArray)
24     mergeSortHelper(auxiliaryArray, middleIdx+1, endIdx, mainArray)
25     doMerge(mainArray, startIdx, middleIdx, endIdx, auxiliaryArray)
26 }
27
28 func doMerge(mainArray []int, startIdx, middleIdx, endIdx int, auxilia
29     k := startIdx
30     i := startIdx
31     j := middleIdx + 1
32     for i <= middleIdx && j <= endIdx {
33         if auxiliaryArray[i] <= auxiliaryArray[j] {
```

Solution 1Solution 2Solution 3

```
1 package main
2
3 func MergeSort(array []int) []int {
4     // Write your code here.
5     return nil
6 }
7
```

Our Tests

Custom Output

Submit Code

```
18 def is_prime(n):
19     if n < 2:
20         return False
21     if n == 2:
22         return True
23     if n % 2 == 0:
24         return False
25     for i in range(3, int(n**0.5) + 1, 2):
26         if n % i == 0:
27             return False
28     return True
29
30 def is_prime(n):
31     if n < 2:
32         return False
33     if n == 2:
34         return True
35     if n % 2 == 0:
36         return False
37     for i in range(3, int(n**0.5) + 1, 2):
38         if n % i == 0:
39             return False
40     return True
41
42 def is_prime(n):
43     if n < 2:
44         return False
45     if n == 2:
46         return True
47     if n % 2 == 0:
48         return False
49     for i in range(3, int(n**0.5) + 1, 2):
50         if n % i == 0:
51             return False
52     return True
```

Run or submit code when you're ready.