Our Solution(s) Run Code

Solution 1 Solution 2

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    class LinkedList {
        var value: Int
        var next: LinkedList?

        init(value: Int) {
            self.value = value
        }
    }

    // O(n + m) time | O(1) space - where n is the number of nodes in
    // Linked List and m is the number of nodes in the second Linked L
    func mergeLinkedLists(_ headOne: LinkedList, _ headTwo: LinkedList
        recursiveMerge(headOne, headTwo, nil)
        if headOne.value < headTwo.value {
            return headOne
        }
        return headTwo
    }

    func recursiveMerge(_ p1: LinkedList?, _ p2: LinkedList?, _ p1Prev
        if p1 == nil {
            p1Prev!.next = p2
            return
        }

        if p2 == nil {
            return
        }

        if p1!.value < p2!.value {
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```swift
class Program {
    // This is an input class. Do not edit.
    class LinkedList {
        var value: Int
        var next: LinkedList?

        init(value: Int) {
            self.value = value
        }
    }

    func mergeLinkedLists(_ headOne: LinkedList, _ headTwo: LinkedList
        // Write your code here.
        return headOne // replace me
    }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.