Our Solution(s) Run Code

Solution 1 Solution 2

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

package main

type LinkedList struct {
  Value int
  Next  *LinkedList
}

// O(n + m) time | O(1) space - where n is the number of nodes in the
// Linked List and m is the number of nodes in the second Linked List
func MergeLinkedLists(headOne *LinkedList, headTwo *LinkedList) *Link
  p1 := headOne
  var p1Prev *LinkedList
  p2 := headTwo
  for p1 != nil && p2 != nil {
    if p1.Value < p2.Value {
      p1Prev = p1
      p1 = p1.Next
    } else {
      if p1Prev != nil {
        p1Prev.Next = p2
      }
      p1Prev = p2
      p2 = p2.Next
      p1Prev.Next = p1
    }
  }

  if p1 == nil {
    p1Prev.Next = p2
  }

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

// This is an input struct. Do not edit.
type LinkedList struct {
  Value int
  Next  *LinkedList
}

func MergeLinkedLists(headOne *LinkedList, headTwo *LinkedList) *Linke
  // Write your code here.
  return nil
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.