

PromptScratchpadOur Solution(s)Video ExplanationRun Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 type LinkedList struct {
6     Value int
7     Next *LinkedList
8 }
9
10 // O(n + m) time | O(n + m) space - where n is the number of nodes in the first
11 // Linked List and m is the number of nodes in the second Linked List
12 func MergeLinkedLists(headOne *LinkedList, headTwo *LinkedList) *LinkedList {
13     recursiveMerge(headOne, headTwo, nil)
14     if headOne.Value < headTwo.Value {
15         return headOne
16     }
17     return headTwo
18 }
19
20 func recursiveMerge(p1, p2, p1Prev *LinkedList) {
21     if p1 == nil {
22         p1Prev.Next = p2
23         return
24     }
25     if p2 == nil {
26         return
27     }
28
29     if p1.Value < p2.Value {
30         recursiveMerge(p1.Next, p2, p1)
31         return
32     }
33
34     if p1Prev != nil {
35         p1Prev.Next = p2
36     }
37     newP2 := p2.Next
38     p2.Next = p1
39     recursiveMerge(p1, newP2, p2)
40 }
41
```