

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     public static class LinkedList {
5         int value;
6         LinkedList next;
7
8         LinkedList(int value) {
9             this.value = value;
10            this.next = null;
11        }
12    }
13
14    // O(n + m) time | O(n + m) space - where n is the number of nodes in the first
15    // Linked List and m is the number of nodes in the second Linked List
16    public static LinkedList mergeLinkedLists(LinkedList headOne, LinkedList headTwo) {
17        recursiveMerge(headOne, headTwo, null);
18        return headOne.value < headTwo.value ? headOne : headTwo;
19    }
20
21    public static void recursiveMerge(LinkedList p1, LinkedList p2, LinkedList p1Prev) {
22        if (p1 == null) {
23            p1Prev.next = p2;
24            return;
25        }
26        if (p2 == null) return;
27
28        if (p1.value < p2.value) {
29            recursiveMerge(p1.next, p2, p1);
30        } else {
31            if (p1Prev != null) p1Prev.next = p2;
32            LinkedList newP2 = p2.next;
33            p2.next = p1;
34            recursiveMerge(p1, newP2, p2);
35        }
36    }
37 }
38
```