

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class LinkedList {
4   constructor(value) {
5     this.value = value;
6     this.next = null;
7   }
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 function mergeLinkedLists(headOne, headTwo) {
13   recursiveMerge(headOne, headTwo, null);
14   return headOne.value < headTwo.value ? headOne : headTwo;
15 }
16
17 function recursiveMerge(p1, p2, p1Prev) {
18   if (p1 === null) {
19     p1Prev.next = p2;
20     return;
21   }
22   if (p2 === null) return;
23
24   if (p1.value < p2.value) {
25     recursiveMerge(p1.next, p2, p1);
26   } else {
27     if (p1Prev !== null) p1Prev.next = p2;
28     const newP2 = p2.next;
29     p2.next = p1;
30     recursiveMerge(p1, newP2, p2);
31   }
32 }
33
34 exports.LinkedList = LinkedList;
35 exports.mergeLinkedLists = mergeLinkedLists;
36
```