

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 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(1) space - where n is the number of nodes in the
11 // Linked List and m is the number of nodes in the second Linked List
12 func MergeLinkedLists(headOne *LinkedList, headTwo *LinkedList) *LinkedList {
13     p1 := headOne
14     var p1Prev *LinkedList
15     p2 := headTwo
16     for p1 != nil && p2 != nil {
17         if p1.Value < p2.Value {
18             p1Prev = p1
19             p1 = p1.Next
20         } else {
21             if p1Prev != nil {
22                 p1Prev.Next = p2
23             }
24             p1Prev = p2
25             p2 = p2.Next
26             p1Prev.Next = p1
27         }
28     }
29
30     if p1 == nil {
31         p1Prev.Next = p2
32     }
33 }
```

Solution 1

Solution 2

Solution 3

```
1 package main
2
3 // This is an input struct. Do not edit.
4 type LinkedList struct {
5     Value int
6     Next *LinkedList
7 }
8
9 func MergeLinkedLists(headOne *LinkedList, headTwo *LinkedList) *LinkedList {
10     // Write your code here.
11     return nil
12 }
13
```

Our Tests

Custom Output

Submit Code

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

```

10  x = RandomWalk(1000)
11  x.plot()
12  x = x - x.min()
13  x = x / x.max()
14  x = x * 100
15  x = x + 50
16  x = x * 1.05
17  x = x * 1.05
18  x = x * 1.05
19  x = x * 1.05
20  x = x * 1.05
21  x = x * 1.05
22  x = x * 1.05
23  x = x * 1.05
24  x = x * 1.05
25  x = x * 1.05
26  x = x * 1.05
27  x = x * 1.05
28  x = x * 1.05
29  x = x * 1.05
30  x = x * 1.05
31  x = x * 1.05
32  x = x * 1.05
33  x = x * 1.05
34  x = x * 1.05
35  x = x * 1.05
36  x = x * 1.05
37  x = x * 1.05
38  x = x * 1.05
39  x = x * 1.05
40  x = x * 1.05
41  x = x * 1.05
42  x = x * 1.05
43  x = x * 1.05
44  x = x * 1.05
45  x = x * 1.05
46  x = x * 1.05
47  x = x * 1.05
48  x = x * 1.05
49  x = x * 1.05
50  x = x * 1.05
51  x = x * 1.05
52  x = x * 1.05
53  x = x * 1.05
54  x = x * 1.05
55  x = x * 1.05
56  x = x * 1.05
57  x = x * 1.05
58  x = x * 1.05
59  x = x * 1.05
60  x = x * 1.05
61  x = x * 1.05
62  x = x * 1.05
63  x = x * 1.05
64  x = x * 1.05
65  x = x * 1.05
66  x = x * 1.05
67  x = x * 1.05
68  x = x * 1.05
69  x = x * 1.05
70  x = x * 1.05
71  x = x * 1.05
72  x = x * 1.05
73  x = x * 1.05
74  x = x * 1.05
75  x = x * 1.05
76  x = x * 1.05
77  x = x * 1.05
78  x = x * 1.05
79  x = x * 1.05
80  x = x * 1.05
81  x = x * 1.05
82  x = x * 1.05
83  x = x * 1.05
84  x = x * 1.05
85  x = x * 1.05
86  x = x * 1.05
87  x = x * 1.05
88  x = x * 1.05
89  x = x * 1.05
90  x = x * 1.05
91  x = x * 1.05
92  x = x * 1.05
93  x = x * 1.05
94  x = x * 1.05
95  x = x * 1.05
96  x = x * 1.05
97  x = x * 1.05
98  x = x * 1.05
99  x = x * 1.05
100 x = x * 1.05

```

Run or submit code when you're ready.