

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Unit4
8 {
9     class ForkNodeUtils
10    {
11        public static void CreateForkList<T>(Node<T> ls1, Node<T> ls2, int n)
12        {
13            while(ls2.GetNext() != null)
14                ls2 = ls2.GetNext();
15
16            for (int i = 0; i < n-1; i++)
17                ls1 = ls1.GetNext();
18
19            ls2.SetNext(ls1);
20        }
21
22        public static Node<T> GetForkMeeting<T>(Node<T> ls1, Node<T> ls2)
23        {
24            int size_ls1 = NodeUtils.CountList(ls1);
25            int size_ls2 = NodeUtils.CountList(ls2);
26
27            int diff = Math.Abs(size_ls2 - size_ls1);
28
29            if (size_ls1 > size_ls2)
30            {
31                for (int i = 0; i < diff; i++)
32                    ls1 = ls1.GetNext();
33            }
34
35            else
36            {
37                for (int i = 0; i < diff; i++)
38                    ls2 = ls2.GetNext();
39            }
40
41            while(ls1 != ls2 && ls1 != null && ls2 != null)
42            {
43                ls1 = ls1.GetNext();
44                ls2 = ls2.GetNext();
45            }
46
47            if (ls1 == null || ls2 == null)
48                return null;
```

```
49
50         return ls1;
51     }
52
53     public static Node<T> MergeFrokList<T>(Node<T> l1, Node<T> l2)
54     {
55         Node<T> merge = GetForkMeeting(l1, l2);
56
57         int size_l1 = NodeUtils.CountList(l1);
58         int size_l2 = NodeUtils.CountList(l2);
59
60         Node<T> head = size_l1 > size_l2 ? l1 : l2;
61         Node<T> pos = head;
62
63         while (pos.GetNext() != null)
64             pos = pos.GetNext();
65
66         Node<T> conn = size_l1 > size_l2 ? l2 : l1;
67
68         pos.SetNext(conn);
69
70         while (conn.GetNext() != merge)
71             conn = conn.GetNext();
72
73         conn.SetNext(null);
74
75
76         return head;
77     }
78
79 }
80 }
81
```