

# LCS.java

```

1 // Creates a linked list used in the chaining method
2 public class LCS {
3
4     public LCS(){
5
6     }
7
8     public String callLCS(String X, String Y) {
9         int XLen = X.length();
10        int YLen = Y.length();
11        if(XLen == 0 || YLen == 0)
12        {
13            return "";
14        }
15        else if (X.charAt(XLen-1) == Y.charAt(YLen-1))
16        {
17            return X.charAt(XLen-1) + callLCS(X.substring(0, XLen-1),Y.substring(0,
18            YLen-1));
19        }
20        else
21        {
22            int subXLen = callLCS(X.substring(0, XLen-1),Y).length();
23            int subYLen = callLCS(X,Y.substring(0, YLen-1)).length();
24
25            if (subXLen>subYLen)
26            {
27                return callLCS(X.substring(0, XLen-1),Y);
28            }
29            else
30            {
31                return callLCS(X,Y.substring(0, YLen-1));
32            }
33        }
34    }
35    public String callLCS2(String X, String Y) {
36        int XLen = X.length();
37        int YLen = Y.length();
38        int[][] arr = new int[XLen + 1][YLen + 1];
39
40        for (int i = XLen - 1; i >= 0; i--)
41        {
42            for (int j = YLen - 1; j >= 0; j--)
43            {
44                if (X.charAt(i) == Y.charAt(j))
45                    arr[i][j] = arr[i + 1][j + 1] + 1;
46                else
47                    arr[i][j] = Math.max(arr[i + 1][j], arr[i][j + 1]);
48            }
49        }
50        int i = 0, j = 0;
51        String sb = "";
52        while (i < XLen && j < YLen)
53        {
54            if (X.charAt(i) == Y.charAt(j))
55            {
56                sb= sb+ X.charAt(i);
57                i++;
58                j++;
59            }
60        }
61    }
62 }

```

LCS.java

```
59         else if (arr[i + 1][j] >= arr[i][j + 1])
60             i++;
61         else
62             j++;
63     }
64     return sb;
65 }
66 }
67
```