

Longest Common Sequence(最长公共序列)

注意区分最长公共子串和最长公共序列。。。

问题:

给出两个字符串或者字符数组，找出其中的最长公共序列的长度以及输出所有可能的最长公共序列。

$X = [1...m] = \{A,B,C,B,D,A,B\}$
 $Y = [1...n] = \{B,D,C,A,B,A\}$
输出的结果是4，以及“BDAB”、“BCAB”、“BCBA”。

方案：动态规划

本题和计算两字串之间的Lev距离类似，都是利用二维动态数组计算距离，只不过这次是相同的距离。因此可以按照DTW方法求出两个字符串之间的距离。

直接看代码吧:

```
1. //先求出最长公共序列的长度
2. public int getLCSlength(String m, String n) {
3.     m = "*" + m;
4.     n = "*" + n;//保证第一个字符相同
5.     char[] arr1 = m.toCharArray();
6.     char[] arr2 = n.toCharArray();
7.     int l1 = arr1.length;
8.     int l2 = arr2.length;
9.     return lcs(arr1, arr2, l1, l2);
10. }
11.
12. private int lcs(char[] arr1, char[] arr2, int l1, int l2) {
13.     int[][] dp = new int[l1][l2];
14.     //初始化
15.     for (int i = 0; i < l1; i++) dp[i][0] = 0;
16.     for (int j = 0; j < l2; j++) dp[0][j] = 0;
17.
18.     for (int i = 1; i < l1; i++) {
19.         for (int j = 1; j < l2; j++) {
20.             if (arr1[i] == arr2[j]) {
21.                 dp[i][j] = dp[i-1][j-1] + 1;
22.             } else {
23.                 dp[i][j] = Math.max(dp[i-1][j], dp[i][j-1]);
24.             }
25.         }
26.     }
27.     return dp[l1-1][l2-1];
28. }
```

在eclipse上通过测试。

接下来把可能的最长公共子串全部输出来。

方法回溯法。这种方法其实就是上述DTW的倒着表示。

- 如果arr[i] == arr[j], 则把当前字符放入字符串，然后i-,j-;
- 否则的话就是转到max(dp[i-1][j], dp[i][j-1])
- 接着第二条，如果dp[i-1][j] ==dp[i][j-1],那么就存在多个结果。

看代码：本代码接着上面，因为必须把dp矩阵求解出来才能继续回溯。

```
1. public void backTrace(char[] arr1, char[] arr2, int[][] dp, int l1, int l2, List<String> ret, String cur) {
2.     while (l1 > 0 && l2 > 0) { //因为之前我们把字符串之前加了一个相同字符
3.         if (arr1[l1] == arr2[l2]) {
4.             cur += arr1[l1];
5.             l1--;
6.             l2--;
7.         } else {
8.             if (dp[l1-1][l2] == dp[l1][l2-1]) {
9.                 backTrace(arr1, arr2, dp, l1-1, l2, ret, cur);
10.                backTrace(arr1, arr2, dp, l1, l2-1, ret, cur);
11.                return;
12.            } else {
13.                dp[l1-1][l2] > dp[l1][l2-1]?l1--:l2--;
14.            }
15.        }
16.    }
17.    ret.add(cur);//这里还需要反转。
18. }
```

eclipse测试通过