**最长公共子序列问题实验报告**

1. **实验目的**

通过实践，加深对动态规划问题求解方法和最长公共子序列问题的理解，提高编程能力。

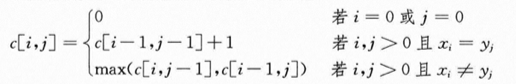
1. **实验内容**
2. 寻找两个字符串的最长公共子序列LCS。
3. 如果存在LCS，则打印出LCS。
4. **实验步骤**

**（1）寻找LCS问题的最优子结构**

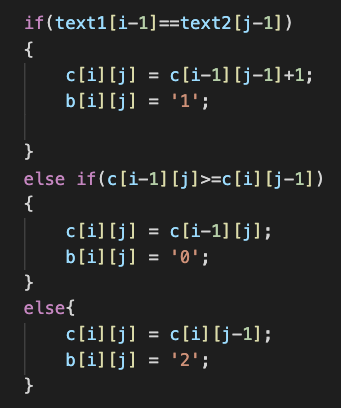
令 X = <x1, x2, …, xm> 和 Y = <y1, y2, …, yn>为两个序列，Z = <z1,z2, ... , zk >为X和Y的任意LCS。

1. 如果xm = yn, 则zk = xm = yn 且Zk-1是Xm-1和Yn-1的一个LCS。
2. 如果xm yn, 则zk xm 意味着Z是是Xm-1和Y的一个LCS。
3. 如果xm yn, 则zk yn 意味着Z是X和Yn-1的一个LCS。

根据上述最优子结构，可得如下公式：

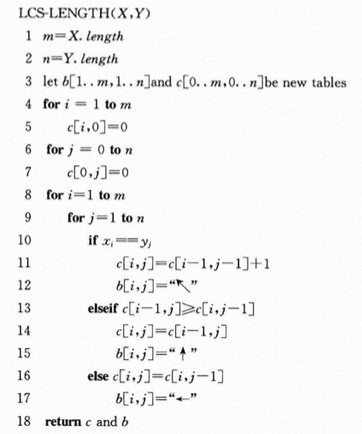
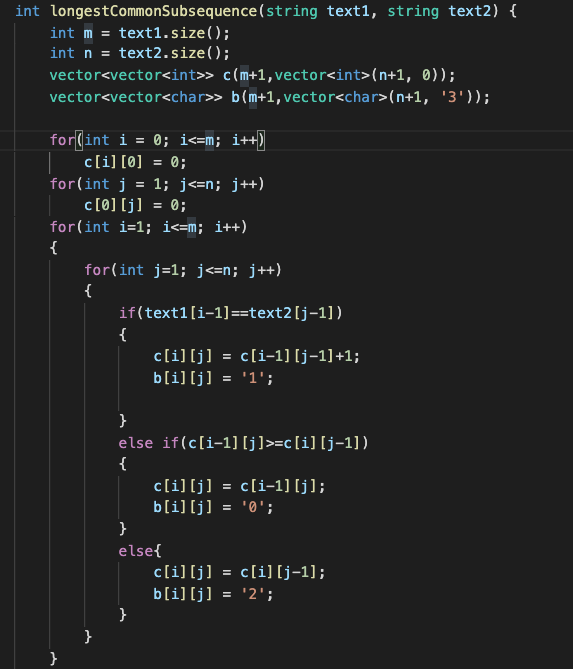


c[i , j]表示Xi和Yj的LCS的长度。



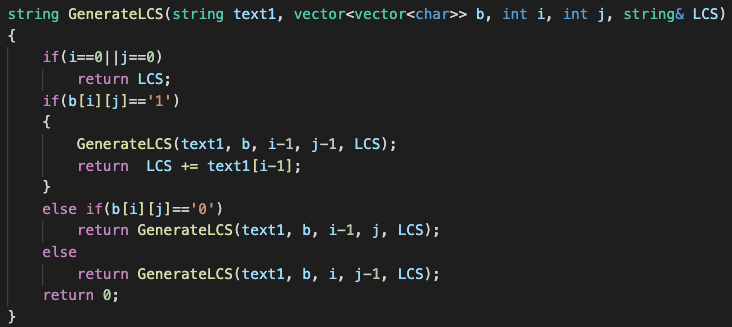
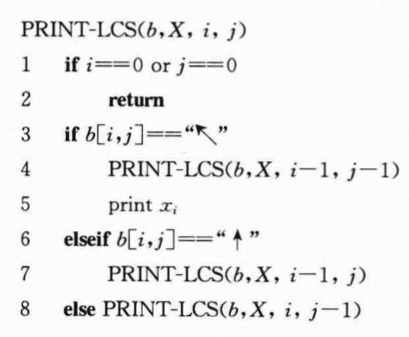
**（2）算法过程描述**

对于X = <x1, x2, …, xm> 和 Y = <y1, y2, …, yn>这两个序列，设立一张表c[0..m, 0..n]，其中c[i , j]表示Xi和Yj的LCS的长度。并且按照行优先的顺序计算表项。过程还维护一个表b[1..m , 1..n] 用于帮助构造LCS， b[i, j]指向的表项对应计算c[i, j]时所选择的子问题最优解。过程返回表b和表c, c[m, n]保存了X和Y的LCS的长度。下面是动态规划过程的伪码和对应的C++实现。

****

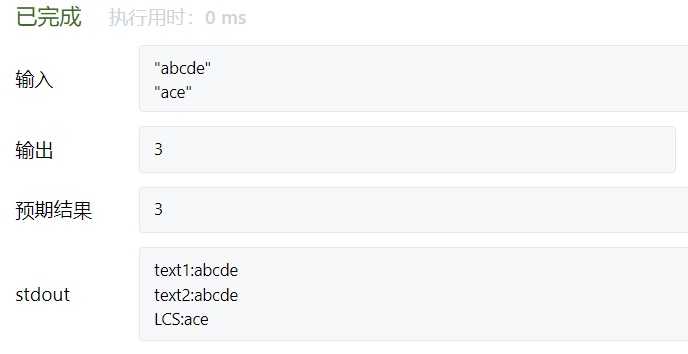
1. **构造LCS**

我们可以用动态规划过程中返回的表b快速构造X和Y的LCS，只需简单地从b[m, n]开始，并按箭头方向追踪下去即可。当在表项中遇到“1”时，意味着xi = yi是LCS的一个元素。按照这种方法，就可以按照逆序一次构造出LCS的所有元素。下面是生成LCS的伪码和C++实现，在实现过程中，用“0”代替截屏2021-12-22 下午2.42.30，用‘“1”代替**截屏2021-12-22 下午2.43.58**，用“2”代替**截屏2021-12-22 下午2.44.37。**

****

1. **实验结果**

测试1：



成功找到LCS

测试2：



未找到LCS

**五、实验心得**

对于动态规划问题的求解，最难的部分就在于刻画最优子结构，然而最优子结构可能一开始不是很容易刻画出来，可以首先尝试用朴素的思想去求解问题，在求解过程中一般可以发现问题的最优子结构。