Python算法之旅（第9期）

上期回顾：

描述：石头剪刀布游戏规则：石头打剪刀，布包石头，剪刀剪布。

现在，需要你写一个程序来判断石头剪子布游戏的结果。

函数名：rock\_paper\_scissors(s1, s2):

参数表：s1,s2-- 字符串，分别表示甲乙两个人的出拳，只可能取值在{"石头","剪刀","布"}中。

返回值：字符串，存储了游戏结果，只可能取值在{"甲胜","乙胜","平局"}中。

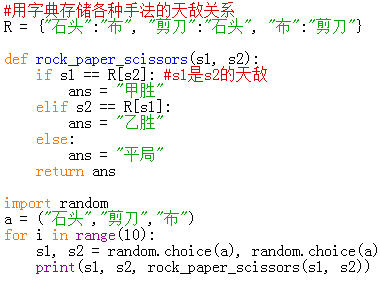
示例1：s1="石头"，s2="剪刀"，返回"甲胜"；

示例2：s1="石头"，s2="布"，返回"乙胜"；

示例3：s1="石头"，s2="石头"，返回"平局"。

算法分析：

解决本题的方法很多，使用字典存储各种手法的天敌关系，然后直接比较字符串和它在字典中的值，无疑是最为简洁的方法。代码如下：



字符串是编程中常用的一种数据结构，也是高考的重点内容，无论是高考还是各种联考，模拟考中都出现了大量与字符串相关的问题，今天我们来看其中一道有趣的题目。

题目：最长连续公共子串（2018 学年第一学期浙江七彩阳光联盟第二次联考17题）

难度：3星 有趣：3星 有用：3星

分类：字符串，穷举，动态规划

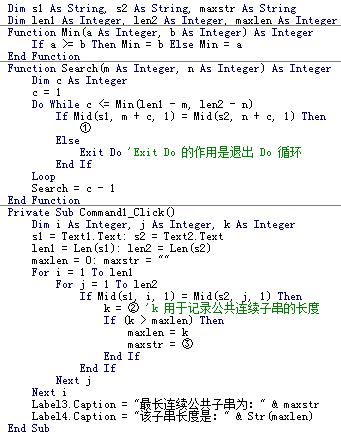
描述：用 VB 编程求两个字符串的最长连续公共子串，程序功能是：在文本框 Text1 和 Text2分别输入任意两串字符串 s1 和 s2，单击命令按钮 Command1，程序在标签 Label3 和 Label4 中分别输出这两个子串的最长连续公共子串和子串的长度。

其算法思想是：分别从字符串 s1 和 s2 的左边第一个字符开始检查，若发现这两个字符中有一个字符相同，则以这个字符为基准向右边扩大搜索范围，先检查其后面的第一个字符是否相同，若还是相同则继续搜索，直到找到不同的字符为止。然后按照该方法依次继续往后搜索，直到查找结束。

程序界面如图所示，实现上述功能的 VB 程序如下，请问答下列问题：

（1）根据代码，若输入的 s1 为“Teacher”，s2 为“teaciher”，则最长连续公共子串为\_\_\_\_\_\_\_\_。

（2）请在划线处填入合适代码。



答案：（1）eac （1分）

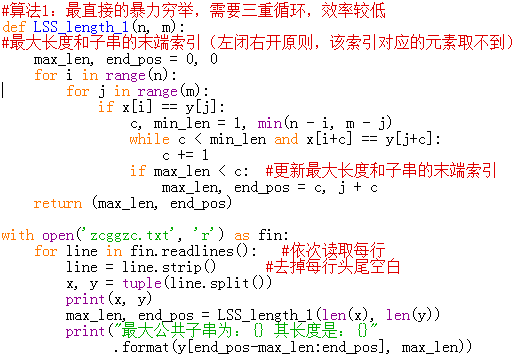
（2）①c = c + 1 （1分）

②Search(i, j) + 1或 Search(i-1, j-1)（2分）

③Mid(s1, i, maxlen) 或Mid(s1, i, k)

或Mid(s2, j, k) 或mid(s2,j,maxlen) （2分）

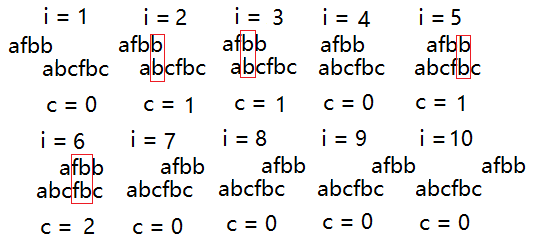
对应Python代码：

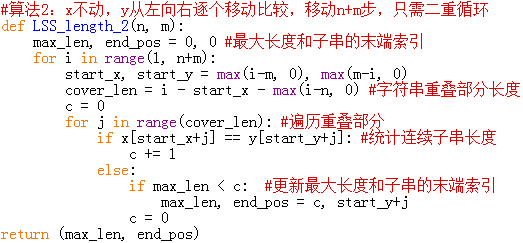


算法1是最直接的暴力穷举，需要三重循环，效率较低；我们可以对其进行改进，使用一个优化了的穷举法，只需二重循环，效率有所提高。其原理如下：

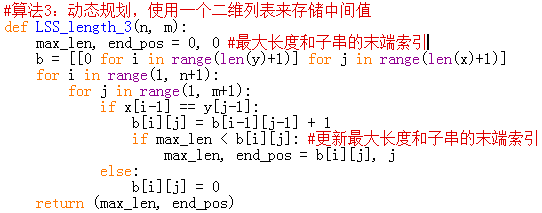
将字符串x和y分别写在两把直尺上面，然后将x固定，x的左端和y的右端对齐，然后向右移动y，比较重叠部分字符串中公共子串的长度，直到y移动到x的右端。在这个过程中求得的最大公共子串就是x和y的最大公共子串。

     下图是求解过程的图示，红框表示重叠部分的公共子串。





前面两种算法都属于穷举法，他们的特点是算法思路简洁，便于编程，但是时间复杂度较高，属于低效算法。我们可以用动态规划思想解决“最长连续公共子串”问题，它的特点是引进一个二维列表b，用b[i][j]记录x[i]与y[j]的连续公共子串的长度，最长公共子串的长度为max(b[i][j]), i∈[1,n],j∈[1,m]。代码如下：



至此，我们为大家展示了解决“最长连续公共子串”问题的几种常用算法，简明高效的算法与简洁明了的Python语言相结合，是学习算法，解决实际问题的有力武器。

课后思考：

考虑到算法3使用二维列表存储中间值，需要较大的空间，并且存在浪费现象，我们还可以对其进行降维优化，使用2个一维列表来存储中间值。

聪明的你不妨想一想，怎样来写出这段降维优化的代码呢？

另外，如果你有更 Pythonic（优雅的、地道的、整洁的）代码，或者与本文不同的算法思路和代码实现，请你一定留言或联系我，让我们一起讨论，共同进步。