

算法设计与分析第二次上机：寻找最长公共子序列

学号：21009200158 姓名：游霄童

时间：2023/10/24

代码语言为Python

问题描述:

给定两个字符串 `text1` 和 `text2`，返回这两个字符串的最长 **公共子序列** 的长度与该 **公共子序列**。如果不存在 **公共子序列**，返回 0。

一个字符串的 **子序列** 是指这样一个新的字符串：它是由原字符串在不改变字符的相对顺序的情况下删除某些字符（也可以不删除任何字符）后组成的新字符串。

源代码如下：：

```
# 找出最长公共子序列
# Created by 游霄童 2023/10/24

# 计算长度
def lcs(str1, str2, dp):
    len1 = len(str1)
    len2 = len(str2)

    for i in range(1, len1 + 1): # dp表第一行和第一列元素为0，所以i和j要从1开始，到最后一个元素len1+1
        for j in range(1, len2 + 1):
            if str1[i - 1] == str2[j - 1]: # i=1时字符串从a[0]开始
                dp[i][j] = dp[i - 1][j - 1] + 1
            else:
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1])

    return dp[len1][len2] # dp表右下角最后一个元素为最长公共子序列长度

# 获得具体的子序列
def getlcs(str1, str2, dp):
    i = len(str1)
    j = len(str2)
    res = ""

    while (i != 0 and j != 0): # 两个字符串最后一个元素相等的话，选择一个字符串中的元素添加到res="" 中
        if (str1[i - 1] == str2[j - 1]):
```

```

        res += str1[i - 1]
        i -= 1
        j -= 1
    else:
        if (dp[i][j] == dp[i - 1][j]): # dp[i][j]从左边来还是从上边来
            i -= 1
        else:
            j -= 1

    return res[::-1] # res是从右往左的字符串，所以要逆序将其输出为从左往右的字符串

str1 = input("请输入序列1: ")
str2 = input("请输入序列2: ")

lenA = len(str1)
lenB = len(str2)

dp = [[0 for i in range(lenB + 1)] for j in range(lenA + 1)] # 生成一个行为lenB+1, 宽为lenA+1的二维数组

length = lcs(str1, str2, dp)
print("最长公共子序列长度为: ", length)

res = getlcs(str1, str2, dp)
print("最长公共子序列为: ", res)

```

测试用例结果如下图所示：

测试用例1

- adsdsakjjjj
- sdakjjkkjjj

```

D:\CONDA\python.exe D:\游霄童\2023\20231024_算法设计与分析第二次上机\Longest.py
请输入序列1: adsdsakjjjj
请输入序列2: sdakjjkkjjj
最长公共子序列长度为: 8
最长公共子序列为: sdakjjjj

进程已结束，退出代码为 0

```

测试用例2(特殊情况)

- dsad
- kolkllkk

```
D:\CONDA\python.exe D:\游霄童\2023\20231024_算法设计与分析第二次上机\Longest.py
请输入序列1: dsad
请输入序列2: kolklkk
最长公共子序列长度为: 0
最长公共子序列为:

进程已结束，退出代码为 0
```

测试用例3

- dsahjukhnaskhcioasj211564856196854
- sdajklhcnuxzahiwqhdu8w156nksanh48

```
D:\CONDA\python.exe D:\游霄童\2023\20231024_算法设计与分析第二次上机\Longest.py
请输入序列1: dsahjukhnaskhcioasj211564856196854
请输入序列2: sdajklhcnuxzahiwqhdu8w156nksanh48
最长公共子序列长度为: 15
最长公共子序列为: dajkhnaio15648

进程已结束，退出代码为 0
```

时间复杂度分析

时间复杂度的计算基于两个嵌套循环的次数，其中外部循环运行 $len1$ 次，内部循环运行 $len2$ 次，其中 $len1$ 和 $len2$ 分别是输入字符串 $str1$ 和 $str2$ 的长度。

```
# 计算长度
def lcs(str1, str2, dp):
    len1 = len(str1)
    len2 = len(str2)

    for i in range(1, len1 + 1): # dp表第一行和第一列元素为0，所以i和j要从1开始，到最后一个元素len1+1
        for j in range(1, len2 + 1):
            if str1[i - 1] == str2[j - 1]: # i=1时字符串从a[0]开始
                dp[i][j] = dp[i - 1][j - 1] + 1
            else:
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1])

    return dp[len1][len2] # dp表右下角最后一个元素为最长公共子序列长度
```

因此，该算法的时间复杂度为

$$O(len1 * len2)$$

其中 $len1$ 和 $len2$ 分别是输入字符串 $str1$ 和 $str2$ 的长度。