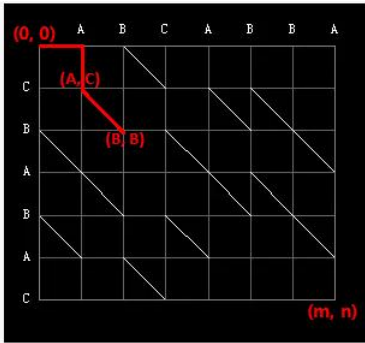


题目描述：

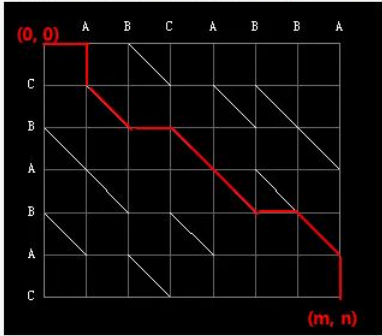
给定两个字符串，分别为字符串 A 与字符串 B。例如 A 字符串为 ABCABBA，B 字符串为 CBABAC 可以得到下图  $m \times n$  的二维数组，定义原点为  $(0, 0)$ ，终点为  $(m, n)$ ，水平与垂直的每一条边距离为 1，映射成坐标系如下图。

从原点  $(0, 0)$  到  $(0, A)$  为水平边，距离为 1，从  $(0, A)$  到  $(A, C)$  为垂直边，距离为 1；假设两个字符串同一位置的两个字符相同则可以作一个斜边，如  $(A, C)$  到  $(B, B)$  最短距离为斜边，距离同样为 1。

作出所有的斜边如下图， $(0, 0)$  到  $(B, B)$  的距离为 1 个水平边 + 1 个垂直边 + 1 个斜边 = 3。



根据定义可知，原点终点的最短距离路径如下图红线标记，最短距离为9：



输入描述：空格分割的两个字符串A与字符串B，字符串不为“空串”，字符格式满足正则规则：[A-Z]，字符串长度 < 10000

输出描述：原点到终点的最短距离

补充说明：

示例 1

输入：

ABC ABC

输出：

3

说明：

示例 2

输入:

ABCABBA CBABAC

输出:

9

说明:

```
import sys
for line in sys.stdin:

    s1 = list(line.split()[0])
    s2 = list(line.split()[1])
    l1 = len(s1)
    l2 = len(s2)  # 行
    dp = [[0 for _ in range(l1 + 1)] for _ in range(l2 + 1)]
    dp[0] = [i for i in range(l1 + 1)]

    for row in range(1, l2 + 1):
        for col in range(l1 + 1):
            if col == 0:
                dp[row][0] = row
                continue
            if s1[col - 1] == s2[row - 1]:
                dp[row][col] = min(dp[row - 1][col], dp[row][col - 1], dp[row -
1][col - 1]) + 1
            else:
                dp[row][col] = min(dp[row - 1][col], dp[row][col - 1]) + 1
    print(dp[l2][l1])
```