

583. Delete Operation for Two Strings

Given two strings `word1` and `word2`, return *the minimum number of **steps** required to make `word1` and `word2` the same*.

In one **step**, you can delete exactly one character in either string.

Example 1:

Input: `word1 = "sea"`, `word2 = "eat"`

Output: 2

Explanation: You need one step to make "sea" to "ea" and another step to make "eat" to "ea".

Example 2:

Input: `word1 = "leetcode"`, `word2 = "etco"`

Output: 4

Constraints:

- `1 <= word1.length, word2.length <= 500`
- `word1` and `word2` consist of only lowercase English letters.

```
class Solution:
    def minDistance(self, s1: str, s2: str) -> int:
        n = len(s1)
        m = len(s2)
        dp = [[0]*(len(s2)+1) for _ in range(len(s1)+1)]
        for i in range(1,n+1):
            for j in range(1,m+1):
                if s1[i-1]==s2[j-1]:
                    dp[i][j] = dp[i-1][j-1]+1
                else:
                    dp[i][j] = max(dp[i-1][j], dp[i][j-1])
        return n-dp[-1][-1]+m-dp[-1][-1]
```