

1312. Minimum Insertion Steps to Make a String Palindrome

Given a string `s`. In one step you can insert any character at any index of the string.

Return *the minimum number of steps* to make `s` palindrome.

A **Palindrome String** is one that reads the same backward as well as forward.

Example 1:

```
Input: s = "zzazz"
Output: 0
Explanation: The string "zzazz" is already palindrome we don't need any insertions.
```

Example 2:

```
Input: s = "mbadm"
Output: 2
Explanation: String can be "mbdadbmb" or "mdbabdm".
```

Example 3:

```
Input: s = "leetcode"
Output: 5
Explanation: Inserting 5 characters the string becomes "leetcodocteel".
```

Example 4:

```
Input: s = "g"
Output: 0
```

Example 5:

```
Input: s = "no"
Output: 1
```

```
class Solution:
    def minInsertions(self, s: str) -> int:
```

```
if s == s[::-1] or len(s)==1:
    return 0
t = s[::-1]
dp = [[0]*(len(s)+1) for _ in range(len(t)+1)]
for i in range(1,len(s)+1):
    for j in range(1,len(t)+1):
        if s[i-1]==t[j-1]:
            dp[i][j] = 1+dp[i-1][j-1]
        else:
            dp[i][j] = max(dp[i-1][j],dp[i][j-1])
# print(dp)
return len(s)-dp[-1][-1]
```