小码哥教育 SEEMYGO

72. 编辑距离

给定两个单词 *word1* 和 *word2*，计算出将 *word1* 转换成 *word2* 所使用的最少操作数 。

你可以对一个单词进行如下三种操作：

1. 插入一个字符
2. 删除一个字符
3. 替换一个字符

```
输入：word1 = "horse", word2 = "ros"
输出：3
解释：
horse -> rorse (将 'h' 替换为 'r')
rorse -> rose (删除 'r')
rose -> ros (删除 'e')
```

编辑距离算法被数据科学家广泛应用，是用作机器翻译和语音识别评价标准的基本算法。

				s2[0]	s2[1]	s2[2]	s2[3]	s2[4]
			j	a	r	i	s	e
		dp	0	1	2	3	4	5
	i	0						
s1[0]	m	1						
s1[1]	i	2						
s1[2]	c	3						
s1[3]	e	4						

- 假设字符串1（“mice”）为s1，它的长度为n1；字符串2（“arise”）为s2，它的长度为n2
- dp是大小为(n1 + 1) * (n2 + 1)的二维数组
- dp[i][j]是s1[0, i)转换成s2[0, j)的最少操作数
  - s1[0, i)是由s1的前i个字符组成的子串
  - s2[0, j)是由s2的前j个字符组成的子串

- 很显然，dp[n1][n2]就是我们要的答案，就是s1[0, n1)转换成s2[0, n2)的最少操作数
  - 也就是s1转换成s2的最少操作数

				s2[0]	s2[1]	s2[2]	s2[3]	s2[4]
			j	a	r	i	s	e
		dp	0	1	2	3	4	5
	i	0	0	1	2	3	4	5
s1[0]	m	1	1					
s1[1]	i	2	2					
s1[2]	c	3	3					
s1[3]	e	4	4					

- 最左上角的dp[0][0]：代表s1的空子串转换为s2的空子串的最少操作数
  - 其实就是什么也不用做，所以：dp[0][0] = 0

- 第0列的dp[i][0]：代表s1[0, i)转换为s2的空子串的最少操作数
  - 其实就是删除s1[0, i)的所有字符，所以：dp[i][0] = i

- 第0行的dp[0][j]：代表s1的空子串转换为s2[0, j)的最少操作数
  - 其实就是插入s2[0, j)的所有字符，所以：dp[0][j] = j

				0	1	2	3	4
			j	a	r	i	s	e
		dp	0	1	2	3	4	5
	i	0	0	1	2	3	4	5
0	m	1	1	1	2	3	4	5
1	i	2	2	2	2	2	3	4
2	c	3	3	3	3	3	3	4
3	e	4	4	4	4	4	4	3

- 如何求出其他位置的dp[i][j]?
  - dp[i][j]是s1[0, i)转换成s2[0, j)的最少操作数
  - 可以分4种情况讨论

① 先删除s1[0, i)的最后一个字符得到s1[0, i - 1)
- 然后由s1[0, i - 1)转换为s2[0, j)
- 这种情况下，dp[i][j] = 1 + dp[i - 1][j]

② 先由s1[0, i)转换为s2[0, j - 1)，然后在最后插入字符s2[j - 1]，得到s2[0, j)
- 这种情况下，dp[i][j] = dp[i][j - 1] + 1

③ 如果s1[i - 1] != s2[j - 1]，先由s1[0, i - 1)转换为s2[0, j - 1)
- 然后将s1[i - 1]替换为s2[j - 1]，这种情况下，dp[i][j] = dp[i - 1][j - 1] + 1

④ 如果s1[i - 1] == s2[j - 1]，由s1[0, i - 1)转换为s2[0, j - 1)后就不用再做任何操作
- 这种情况下，dp[i][j] = dp[i - 1][j - 1]