<https://leetcode.com/problems/delete-operation-for-two-strings/description/>

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.

**Attempt 1: 2024-09-10**

**Solution 1: Longest Common Subsequence (10 min, just refer exactly same solution from Leetcode 1143)**

class Solution {

    public int minDistance(String word1, String word2) {

        int lcsLen = findLcsLen(word1, word2);

        return word1.length() + word2.length() - 2 \* lcsLen;

    }

    private int findLcsLen(String word1, String word2) {

        int len1 = word1.length();

        int len2 = word2.length();

        int[][] dp = new int[len1 + 1][len2 + 1];

        for(int i = len1 - 1; i >= 0; i--) {

            for(int j = len2 - 1; j >= 0; j--) {

                if(word1.charAt(i) == word2.charAt(j)) {

                    dp[i][j] = dp[i + 1][j + 1] + 1;

                } else {

                    dp[i][j] = Math.max(dp[i + 1][j], dp[i][j + 1]);

                }

            }

        }

        return dp[0][0];

    }

}

**Refer to**

[L712.Minimum ASCII Delete Sum for Two Strings (Ref.L72,L583,L1143)](note://WEBa910883e40bb3571527f8753e2b859be)

[L72.Edit Distance (Ref.L115)](note://998ECECFA22D4E1E9533D9AD624A6345)

[L1143.Longest Common Subsequence (Ref.L516,L583)](note://F9909063BABB4A67B28268E5F3054AED)