**Algoritmo de subsecuencia común más larga (LCS)**

**1.Crear la tabla dp**

-creamos una matriz dp de tamaño , en el cual

-m=longitud de X

-n=longitud de Y

-dp[i][j] guarda la longitud de la LCS entre los primeros i caracteres de X y los ㅤㅤprimeros j de Y.

-La primera fila y primera columna se llenan con ceros (cuando una cadena está vacía, la ㅤ LCS es 0).

**2. Llenar la tabla**

-Para cada posición (i, j):

-Si los caracteres coinciden:  
dp[i][j] = dp[i-1][j-1] + 1  
(añadimos 1 a la LCS anterior, porque encontramos un carácter en común).

-Si no coinciden:  
dp[i][j] = max(dp[i-1][j], dp[i][j-1])  
(elegimos la LCS más larga entre:

-ignorar el carácter actual de X

-o ignorar el carácter actual de Y).

**3. Reconstruir la LCS**

-Empezamos desde la esquina inferior derecha (dp[m][n]).

-Si los caracteres de XXX e YYY en esa posición coinciden → los agregamos a lcs\_str y nos movemos en diagonal (i-1, j-1).

-Si no coinciden:

-Vamos hacia arriba si dp[i-1][j] > dp[i][j-1].

-Vamos hacia la izquierda en caso contrario.

-Repetimos hasta llegar a la primera fila o columna.

-Invertimos la lista lcs\_str para tener la LCS en orden correcto.

**4. Resultado**

-dp[m][n] → longitud de la LCS.

-"".join(lcs\_str) → la subsecuencia común más larga.