

CC3001 Algoritmos y Estructuras de Datos**Profesores:** Nelson Baloian, Patricio Poblete, e Iván Sipirán**Auxiliares:** Valentina Alarcón Yáñez, Samuel Chávez Fierro, Antonia G.

Calvo, Cristián Llull, y Raimundo Lorca Correa

**Auxiliar 4**

11 de abril de 2025

P1. Programación Dinámica: Distancia de Levenshtein

La distancia de Levenshtein es una medida de la diferencia entre dos cadenas de texto. Se calcula considerando el número mínimo de operaciones (borrar, insertar, sustituir) que se necesitan para transformar una cadena en otra.

Ejemplos:

- La distancia de Levenshtein entre “gato” y “pato” es 1, puesto que solo se requiere cambiar la “g” por la “p”.
- La distancia de Levenshtein entre “perro” y “gato” es 4. La respuesta óptima es:
 - Se sustituye la “p” de “perro” por una “g”.
 - Se sustituye la “e” por una “a”
 - Se cambia una “r” por una “t”.
 - Se elimina la otra “r”.

Lo que equivale a realizar 4 operaciones.

La idea de este ejercicio es utilizar programación dinámica para crear una función `levenshtein(texto_1, texto_2)` que calcule la distancia de Levenshtein entre dos cadenas.

Para ello, deberá considerar el uso de una matriz M donde cada casilla $M[i][j]$ represente la distancia de Levenshtein entre los primeros i caracteres de `texto_1` y los primeros j caracteres de `texto_2`.



DP When you have to calculate the same case again



Backtracking into the right future, one step at a time