

# PRACTIA 2 EJERCICIO 2: SIMILITUD

Juan Miguel Herrada Acosta

Juan Miguel Herrada Acosta  
Grado en ingeniería informática. Estructura de datos y algoritmos I

- ¿Cuál es la idea básica de este método de similitud?

La distancia de Levenshtein, distancia entre palabras el cual es el número mínimo de operaciones necesarias para modificar una cada de carácter en otra.

Ejemplo:

Si tenemos palabra1 igual a chico y palabra2 igual a chica cual sería la distancia de Levenshtein para que la primera palabra fuera igual a la segunda.

Paso 1º: Eliminamos el carácter 'o': chic.

Paso 2º: Insertamos el carácter 'a': chica.

La distancia de Levenshtein es igual a 2.

- ¿Cómo utilizaremos esta distancia de edición para recuperar del AVL las n palabras más similares?

Crearíamos una PriorityQueue en la cual añadiríamos las palabras con distancia de Levenshtein como comparador, el cual usaremos después para sacar los valores n-esimos los cuales estarán ordenados en la PriorityQueue por la distancia de Levenshtein comparada con la palabra de la cual queremos sacar las palabras similares.

- ¿Qué pasaría si el diccionario lo implementásemos mediante un BSTree, en lugar de un AVLTree?

El funcionamiento sería el mismo, pero al tener la posibilidad que el árbol este degenerado en el peor de los casos el coste sería  $O(n)$ , siendo esto muy superior al coste de un AVLTree.