

➤ **¿Cuál es la idea básica de este método de similitud?**

Este algoritmo también es conocido como distancia de edición. La similitud entre dos cadenas de texto A y B se basa en el conjunto mínimo de operaciones de edición necesarias para transformar A en B, o viceversa. Hay tres operaciones de edición, las cuales son destrucción, inserción y substitución. Entre más cerca de cero es la distancia de Levenshtein más parecidas son las hileras.

➤ **¿Cómo utilizaremos esta distancia de edición (*edit distance*, como también se le conoce a la distancia de Levenshtein) para recuperar del AVL las n palabras más similares?**

Esta distancia la utilizamos para insertar en una cola de prioridad, que vayan primero las que menos distancia tengan y al final las que más distancia tenga, de esa manera las más similares las tendríamos al principio.

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

Si lo implementamos con un BSTree el árbol quedaría totalmente desequilibrado, porque el diccionario esta ordenado, mi árbol solo tendría rama.