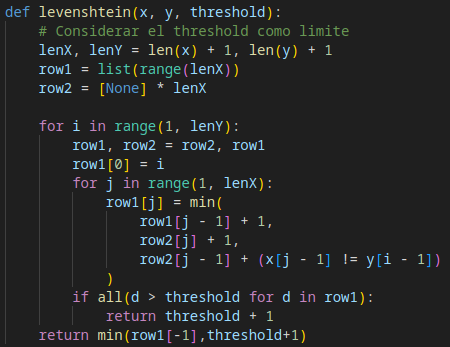


El objetivo de esta función es aplicar el algoritmo de levenshtein con una reducción del coste espacial, dado que solo es necesario tener en memoria las dos últimas filas para poder realizar las comparaciones necesarias para la realización de este algoritmo.

Primero se inicializan las longitudes de la misma manera que en el Levenshtein normal, y inicializamos las dos primeras filas del algoritmo, **row1** como el rango de **lenX** y **row2** como una lista vacía del la misma longitud.

Después comienza el bucle donde compararemos estas 2 filas. Primero las cambiamos para que **row2** sea la base, hacemos que el primer elemento de la fila donde vamos a hacer las comparaciones sea igual a **i**, y iteramos a lo largo del resto de **row1** haciendo las evaluaciones correspondientes a las operaciones de inserción, borrado y sustitución. Guardamos su mínimo y seguimos hasta acabar el bucle, que es cuando devolveremos el último valor obtenido.



Esta función es igual que la de **levenshtein\_reduccion**, excepto que hacemos un uso de **threshold** para dejar de hacer comparaciones en cuanto la distancia de levenshtein se pasa de este **threshold**.

Respecto al código, esto significa hacer una comprobación al final del cálculo de cada fila, de manera que si todos los elementos de la **row1** son estrictamente mayores que **threshold**, entonces dejamos de iterar y devolvemos el **threshold**.