

PROBLEMA 3

En este problema se requiere ordenar de manera externa un archivo de texto de 100 Mb.

Realizamos la resolución utilizando 3 funciones:

La función `contar_lineas_archivo` cuenta la cantidad de datos que tiene el archivo a ordenar. Esto es necesario para poder determinar la cantidad de ciclos de ordenamiento necesarios.

La función `dividir_archivo` divide un archivo de texto grande en dos más pequeños, escribiendo alternadamente `cant_datos` en cada archivo pequeño.

La función `unir_archivo` une dos archivos pequeños en uno más grande, teniendo en cuenta que los archivos pequeños se constituyen de fragmentos ordenados de longitud `cant_datos`.

La función `ordenamiento_inicial` realiza el primer ordenamiento, leyendo de los archivos un bloque de `cant_datos` cargándolos en memoria, ordenándolos y escribiéndolos

alternadamente en los archivos de salida. La cantidad `cant_datos` está fijada en 10.

Por último, la función `ordenar` combina las funciones anteriores para poder ordenar el archivo pasado como parámetro.

Para el test, se hicieron 2 pruebas:

Un test “fino”, que crea una lista pequeña, de 1000 datos, y corre el algoritmo para un archivo de texto donde están esos 1000 datos. El objetivo de este test es asegurar que el algoritmo de ordenamiento externo llega al mismo resultado que la función `sort()` de la lista de python.

Un test “grueso”, que crea un archivo de 5 mb y corrobora que el archivo ordenado resultante tenga la misma cantidad de datos que el archivo original, y que los datos del archivo ordenado estén en orden creciente.