OBSERVACIONES DE LA PRACTICA

Estudiante 1 Cod XXXX

Estudiante 2 Cod XXXX

Estudiante 3 Cod XXXX

# **Preguntas de análisis**

1. ¿Qué modificaciones debería hacer en la librería DISCLib para crear un nuevo algoritmo de ordenamiento?, especifique archivos, rutas de y encabezados de las posibles funciones a implementar.

-Se debe agregar un archivo newsort con a función def newsort en: en Disclib\Algorithms\Sorting C:\Users\valeg\OneDrive\Universidad\U-Segundo Semestre\EDA\LabRecursive-L04-G04\DISClib\Algorithms\Sorting. Con el newsort, se puede cambiar los ordenamientos recursivos a iterativos en Disclib\Algorithms\Sorting en def newsort. En este se puede usar una serie de fors, en donde por cada valor en una lista que contiene a los libros, si son menores o iguales que el pivote, son guardados en una lista, y si son mayores, en otra, y luego juntadas una vez se termine el bucle.

1. ¿Cómo se relaciona la opción 10 del menú en el **view.py** con las funciones principales (opciones 7, 8 y 9) dentro del **model.py**?

-La opción 10 del menú en view.py permite que el usuario seleccióné entre algoritmos recursivos o iterativos para las operaciones de búsqueda y filtrado. En el model.py, hay funciones para ejecutar tanto búsquedas como cálculos de promedio de carácter recursivo e iterativo. Cómo la función recursiveFilterBooksByRating o iterativeFilterBooksByRating. La ejecución dependerá de lo que elija el usuario.

1. ¿Por qué se usa la máscara en las funciones propuestas?, ¿siempre es necesaria?, argumente su respuesta.

Dentro del laboratorio, se utilizó la máscara para la búsqueda recursiva ya que básicamente hace que el algoritmo de búsqueda recursiva se divida en 2 partes: una función principal y una recursiva. La principal (“máscara”) siendo la que establece el contexto general de la búsqueda, inicializa los parámetros que se necesitan y permite que se use de manera más versátil esa función para evitar la repetición de código también. Y en cuanto a la función recursiva es aquella donde se ubica toda la lógica de la búsqueda como tal. Una vez hecha la recursiva, se invoca en la principal. Por otro lado, en cuanto a si siempre es necesaria la máscara, no siempre es un requisito ya que depende del contexto y de la claridad del código, si la implementación de la búsqueda recursiva es simple y fácil de entender sin una "máscara", no es necesaria.

1. ¿Cuál es la causa del error **“RecursionError: maximum recursion Depth exceded”** al ejecutar el código?

El error RecursionError: maximum recursion depth exceeded ocurre cuando una función recursiva se llama a sí misma demasiadas veces, superando el límite máximo de profundidad de recursión, es decir la función entra de un ciclo infinito entonces el programa aborta y saca el error para que no se quede ahí y acabe con los recursos. Para solucionar esto se debe asegurar que exista un caso base dentro de la función recursiva para detener este posible ciclo infinito. También, debido a que se está procesando una estructura de datos grande, puede ocurrir que llame infinitas veces a la función y esto llene el buffer y se salga del programa.

1. ¿Cuál es la causa por la que termina anormalmente el programa?
2. ¿Qué es lo que hacen las modificaciones del **main** al incluir el uso de **threading** en la ejecución del código?, utilice el código Segmento 10 y la documentación oficial de Python para argumentar su respuesta.
3. ¿Qué diferencias existen entre exceder el límite de recursión y la terminación anormal del programa?

Exceder el límite de recursión ocurre cuando una función recursiva se llama repetidamente sin fin lo que puede resultar en un error de "RecursionError". Por otro lado, la terminación anormal del programa sucede cuando este se detiene abruptamente debido a errores como excepciones no manejadas o fallos en el sistema.