OBSERVACIONES DE LA PRACTICA

Samuel Rodríguez, 202310140.

Nicolas Franco, 202021109.

# **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.

Habría que crear un archivo .py en la ruta DISClib>Algorithmss>Sorting. Este deberá tener los siguientes comandos:

import config

from DISClib.ADT import list as lt

assert config

Posteriormente deberá definir una función que reciba una lista y un criterio de ordenamiento. Esta lista implementará el algoritmo sobre la lista aplicando la función de criterio de ordenamiento para posteriormente retornar la lista ordenada.

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 le permite al usuario elegir entre la recursividad o la iteración como formas de resolver las opciones 7,8 y 9, esto únicamente afectando la variable booleana “rec” y en el model con el uso de la mascara se logra este objetivo.

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

Aunque la mascara no siempre es necesaria, en las funciones propuestas se usa como un intermediario entre las peticiones del usuario y las funciones principales, puesto que esta es la principal encargada de manejar la recursión y lo iteración según lo desee el usuario.

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

El error "RecursionError: maximum recursion depth exceeded" se produce cuando una función recursiva se llama a sí misma sin llegar a un caso base que termine la recursión. En otras palabras, la función recursiva nunca alcanza un punto donde deje de llamarse a sí misma, lo que lleva a un bucle infinito de llamadas recursivas y, eventualmente, al desbordamiento de la pila de llamadas, generando este error.

1. ¿Cuál es la causa por la que termina anormalmente el programa?

La terminación anormal del programa puede deberse a varios factores, como un error no manejado (una excepción no capturada), un fallo en la memoria, una operación no válida, entre otros. Estos problemas pueden detener la ejecución del programa antes de que se complete correctamente.

1. ¿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.

El uso de threading en la ejecución del laboratorio ayuda a ejecutar varias funcionalidades de este a la vez, mejorando el rendimiento y permitiendo que haya operaciones que se ejecuten al simultáneamente sin ningún problema pero se debe ser cuidadoso al implementarlo puesto que puede ocasionar errores. Así mismo, en el laboratorio el argumento “target” de las funciones asociadas al threading es “cycle\_menu”, y al iniciar un nuevo thread, **se aumenta el límite de recursión y se reserva la memoria, con lo que se evita el error de la pregunta 4.**

1. ¿Qué diferencias existen entre exceder el límite de recursión y la terminación anormal del programa?

Exceder el límite de recursión sucede cuando una función recursiva se llama a sí misma demasiadas veces, agotando la pila de llamadas y generando un error de recursión. Mientras que la terminación anormal del programa sucede cuando el programa se detiene de manera inesperada debido a un error no manejado, una excepción no capturada u otros problemas como fallos de memoria o operaciones no válidas.