

Laboratorio

Práctica 7. Ordenamiento de datos y GUI

Unidad 7. Búsqueda, ordenamiento y complejidad

1. Objetivos

- Implementar y comparar la velocidad de ejecución diferentes algoritmos de ordenamiento.
- Incorporar funciones de ordenamiento de datos a un programa con interfaz gráfica *Tkinter*.

2. Tareas a realizar

Esta práctica consiste en completar un programa que tiene interfaz gráfica de usuario (GUI, por sus siglas en inglés) y que permite ordenar un conjunto de datos utilizando diferentes algoritmos de ordenamiento.

2.1. Resumen de la aplicación

El programa a completar utiliza la librería *Tkinter* para construir una GUI que le ofrezca al usuario dos alternativas (desde archivo o datos aleatorios) para generar una lista de datos que el programa deberá ordenar. El usuario puede seleccionar también uno de tres algoritmos de ordenamiento (burbuja, selección y el nativo de Python) y observar el reporte de la velocidad de ejecución. Finalmente, la GUI ofrece la alternativa de observar en cámara lenta el proceso de ordenamiento de los datos.

Junto con esta guía se proporcionarán los siguientes archivos para el desarrollo de la práctica:

- `main.py`: contiene el programa principal que configura la GUI junto con otras funciones auxiliares.
- `sortfunc.py`: contiene las funciones de ordenamiento de datos.
- `data.txt`: contiene un ejemplo de archivo con datos de entrada.

2.2. Instalación de librerías

Para la ejecución del código de la práctica es necesario instalar algunas librerías (Python *packages*). Ejecute los siguientes comandos, uno por uno, antes de empezar a probar el programa:

- `sudo apt-get install python3-tk`
- `sudo apt-get install python3-dev`
- `sudo apt-get install python3-numpy`
- `sudo apt-get install python3-matplotlib`

Asegúrese que está conectado a internet antes de ejecutar los comandos y que no haya mensajes de error.

2.3. Partes faltantes del programa

La mayor parte del código del programa ya se encuentra en el archivo `main.py`, sin embargo, algunas partes hacen falta, tanto en `main.py` como en `sortfunc.py`. Las partes faltantes están descritas con comentarios multilínea en ambos archivos fuente y son:

- En la función `animacionBurbuja` es necesario agregar el código que implementa el algoritmo de **ordenamiento por burbuja**. Note que en cada iteración del ciclo externo deberán ir las líneas de código allí indicadas para generar la animación correspondiente.
- De igual manera, en la función `animacionSeleccion` debe agregarse el correspondiente algoritmo de **ordenamiento por selección**.
- En la función `generarLista` debe agregarse el código necesario para **generar una lista** de números aleatorios del tamaño indicado por el usuario.
- En el archivo `sortfunc.py` deben completarse todas las funciones de acuerdo a su especificación.
- Finalmente, cree un tercer archivo fuente que contenga una función que le haga múltiples **pruebas** de caja negra y de caja blanca a las funciones `sortBurbuja` y `sortSeleccion`.

Opcional: agregue dos opciones más de algoritmos de ordenamiento: Merge sort y quick sort.

3. Evaluación

La evaluación se basará en los códigos enviados y una sustentación oral sobre los temas de la práctica. Además, se tendrá en cuenta una bonificación para quien haga la parte opcional.