

Heapsort y estudio experimental

1. Introducción

El objetivo de este laboratorio es el de agregar a la librería de ordenamiento `Sortlib.kt` el algoritmo Heapsort, y hacer una estudio experimental con los algoritmos implementados hasta ahora en la librería.

2. Actividades a realizar

La primera actividad que debe realizar, es la de agregar a la librería de ordenamiento `Sortlib.kt`, el algoritmo Heapsort implementado siguiendo el pseudo código presentado en la página 159 de [1]. El algoritmo Heapsort debe ejecutarse cuando se escoge en la familia de algoritmos `nlgn`, de la opción `-a` de la línea de comando.

Para la segunda actividad debe hacer un estudio experimental. Debe ejecutar las tres familias de algoritmos (`nlgn`, `On2`, y `all`), que vienen dadas con la opción `-a` de la línea de comando. Los algoritmos deben ordenar secuencias de la clase `random` y cada secuencia debe ser ejecutada tres veces, es decir, se debe aplicar la opción `-t 3`. El tamaño de las secuencias del estudio experimental son 10.000, 20.000, 30.0000, 40.000 y 50.0000.

Una vez efectuados los resultados experimentales, debe realizar un breve reporte con los resultados obtenidos y su análisis. El reporte debe contener las tres imágenes generadas por la ejecución de las tres familias de algoritmos (`nlgn`, `On2`, y `all`), y un análisis de las mismas. Dentro de su análisis debe incluir si los resultados obtenidos se corresponden a lo indicado en la teoría. En el reporte se debe indicar los datos de la plataforma donde se ejecutaron los algoritmos: sistema de operación, CPU y memoria RAM del computador, versión del compilador Kotlin utilizada y versión de la JVM utilizada. **El reporte debe estar en formato PDF.**

3. Sobre la implementación

El código encargado de la manipulación de los argumentos de la línea de comandos, debe hacerse en por medio de funciones en Kotlin implementadas por usted, no se puede hacer uso de librerías externas. Su entrega debe contener los archivos `Sortlib.kt`, `runSortlib.sh`, `Makefile`, `Main.kt`, y los archivos necesarios de la librería `libPlotRuntime`. Los archivos necesarios de la librería `libPlotRuntime` son los siguientes: `PlotRuntime.kt`, `SwingJfxDemoFrontendContext.kt` y la carpeta con archivos `.jar lib`. El archivo `Makefile` debe ser capaz de compilar todos los archivos fuentes `.kt`, incluyendo los archivos de la librería `libPlotRuntime`. Se recomienda modificar el `Makefile` de `libPlotRuntime`, incluyendo el resto de los archivos fuentes. El archivo `runSortlib.sh`, es un *script* cuyo único objetivo es la ejecución del cliente de `Main.kt`. Todo el código debe usar la guía de estilo Kotlin indicada en clase. Asimismo, el código debe estar debidamente documentado.

4. Condiciones de entrega

La versión final del código del laboratorio, el reporte, y la declaración de autenticidad firmada, debe estar contenida en un archivo comprimido, con formato *tar.xz*, llamado *LabSem3_X.tar.xz*, donde *X* es el número de carné del estudiante. La entrega del archivo *LabSem4_X.tar.xz* debe hacerse por la plataforma Classroom, antes de las 11:50 pm del día domingo 30 de mayo de 2021.

Referencias

- [1] CORMEN, T., LEIRSESON, C., RIVEST, R., AND STEIN, C. *Introduction to Algorithms*, 3ra ed. McGraw Hill, 2009.

Guillermo Palma / gvpalma@usb.ve / Mayo 2021