

Ordenamiento en tiempo lineal

1. Introducción

El objetivo de este laboratorio es el de agregar a la librería de ordenamiento `Sortlib.kt` los algoritmos de ordenamiento *Counting sort* y *Radix sort*, y hacer un estudio experimental con los algoritmos implementados hasta ahora en la librería.

2. Actividades a realizar

La primera actividad consiste en agregar a la librería `Sortlib.kt` las implementaciones de los algoritmos de ordenamiento *Counting sort* y *Radix sort*:

Counting-sort: presentado en la página 195 de [1].

Radix-sort: introducido en la página 198 de [1].

Para el algoritmo *Radix sort* debe decidir cuál es el algoritmo estable a utilizar.

Estos algoritmos se deben ejecutar junto los algoritmos *Quicksort* e *Introsort*, formando una nueva familia de algoritmos, llamada `ln`, que se ejecutará con la opción `-a` de la línea de comando. También estos algoritmos deben ser agregados en la familia de algoritmos `all`. Es decir, la opción `-a` se puede ejecutar con cuatro familias de algoritmos. En consecuencia, es válido ejecutar como entrada de la línea de comando de `runSortlib.sh`, las opciones `-a nlgn`, `-a On2`, `-a all` y `-a ln`.

La segunda actividad consiste en hacer un estudio experimental con la familia de algoritmos `ln`. 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`. Los tamaños de las secuencias del estudio experimental son 200.000, 400.000, 600.000, 800.000 y 1.000.000.

Una vez efectuados los resultados experimentales, debe realizar un breve reporte con los resultados obtenidos y su análisis. El reporte, **que debe estar en formato PDF**, debe contener los siguientes elementos:

- La imagen generada por la ejecución de la clase de secuencia `random`.
- Detalles de la implementación de los algoritmos *Counting sort* y *Radix sort*.
- Un análisis del resultado mostrado en la gráfica.
- Indicar si el resultados obtenido se corresponde a lo esperado en la teoría.
- Los datos de la plataforma donde se ejecutaron los algoritmos: sistema de operación, modelo de CPU, cantidad de memoria RAM del computador, versión del compilador Kotlin y versión de la JVM utilizada.

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`. 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, deben estar contenidos en un archivo comprimido, con formato *tar.xz*, llamado *LabSem5_X.tar.xz*, donde *X* es el número de carné del estudiante. La entrega del archivo *LabSem5_X.tar.xz* debe hacerse por la plataforma Classroom, antes de las 11:50 pm del día domingo 13 de junio de 2021.

Referencias

- [1] CORMEN, T., LEISERSON, C., RIVEST, R., AND STEIN, C. *Introduction to algorithms*, 3rd ed. MIT press, 2009.