

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

A continuación la descripción de las actividades a realizar.

2.1. Implementación de los algoritmos de ordenamiento lineal

La primera actividad consiste en agregar a la librería `Sortlib.kt` las implementaciones de los siguientes algoritmos:

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

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

Para el algoritmo *Radix sort* debe utilizar como algoritmo estable a *Counting sort*. En `Sortlib.kt` estos algoritmos deben ser implementados como procedimientos que solo reciben como argumento, el arreglo a ordenar. Los nombres a usar en los procedimientos son: `countingSort` y `radixSort`.

2.2. Estudio experimental

La segunda actividad consiste en un estudio experimental que incluye a todos los algoritmos de la librería `Sortlib.kt` que tienen como peor tiempo $O(n * \log n)$, los algoritmos de ordenamientos lineal y las variantes de Quicksort.

Debe ejecutar los algoritmos indicados anteriormente sobre arreglos de enteros generados aleatoriamente. Para cada tamaño de arreglo se debe ejecutar los algoritmos sobre 3 arreglos aleatorios con valores en el intervalo $[0, 1,000,000)$. El tiempo a reportar para cada uno de los algoritmos, en cada tamaño de arreglo, es el tiempo promedio, en segundos, de esos 3 arreglos. Observe que para que el experimento sea válido, cada variante de Quicksort, debe ordenar el mismo arreglo generado aleatoriamente. Los tamaños de los arreglos son: $[1,000,000, 2,000,000, 3,000,000, 4,000,000]$.

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:

1. Portada.
2. 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. Una tabla con los tiempos promedios (con su desviación estándar), para cada uno de los algoritmos, en los cuatro tamaños de arreglos indicados. En la tabla el valor promedio va seguido de la desviación estándar, la cual está en un círculo con el signo \pm adelante. La tabla tiene el siguiente formato:

Nombre del algoritmo	n=1.000.000	n=2.000.000	n=3.000.000	n=4.000.000
----------------------	-------------	-------------	-------------	-------------

4. Un gráfico del tiempo promedio (eje y) versus tamaño del arreglo (eje x), en donde se pueda observar el comportamiento de los algoritmos.
5. Un análisis de los resultados obtenidos.

3. Condiciones de entrega

Para este laboratorio el único código que debe entregar es el de la librería `Sortlib.kt`. La versión final del código del laboratorio, el informe y la declaración de autenticidad firmada, deben estar contenidas en un archivo comprimido, con formato *tar.xz*, llamado *LabSem5_X_Y.tar.xz*, donde *X* y *Y*, son los números de carné de los estudiantes. La entrega del archivo *LabSem5_X_Y.tar.xz*, debe hacerse por la plataforma Classroom, antes de las 11:50 pm del día domingo 11 de junio de 2023.

Referencias

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