

Ordenamiento en tiempo lineal

1. Introducción

El objetivo de este laboratorio es el de agregar a la librería `Ordenamiento.kt` los algoritmos *Counting sort* y *Radix sort*, y hacer un estudio experimental comparativo, con los algoritmos de ordenamiento de 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 `Ordenamiento.kt` las implementaciones de los siguientes algoritmos, con sus códigos respectivos:

cs: Corresponde al algoritmo *Counting sort* presentado en [1].

rs: Se refiere al algoritmo *Radix sort* mostrado en [1].

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

2.2. Estudio experimental

La segunda actividad consiste en un estudio experimental para comparar los algoritmos de ordenamientos en varias pruebas. Las pruebas que debe ejecutar son las siguientes:

- Prueba *random*

```
>./probarA10rd.sh -s random -n 1000000,2000000,3000000,4000000 -t 3 -a ms,mi,hs,qc,qd,rs,cs -g random
```

- Prueba *randomd*

```
>./probarA10rd.sh -s randomd -n 1000000,2000000,3000000,4000000 -t 3 -a ms,mi,hs,qc,qd -g randomd
```

- Prueba *sorted*

```
>./probarA10rd.sh -s sorted -n 1000000,2000000,3000000,4000000 -t 3 -a ms,mi,hs,qd,rs,cs -g sorted
```

Prueba *sortedd*

```
>./probarA10rd.sh -s sortedd -n 1000000,2000000,3000000,4000000 -t 3 -a ms,mi,hs,qd -g sortedd
```

- Prueba *inv*

```
>./probarA10rd.sh -s inv -n 1000000,2000000,3000000,4000000 -t 3 -a ms,mi,hs,qd,rs,cs -g inv
```

- Prueba *zu*

```
>./probarA10rd.sh -s zu -n 1000000,2000000,3000000,4000000 -t 3 -a ms,mi,hs,qd,rs,cs -g zu
```

- Prueba *media*

```
>./probarA10rd.sh -s media -n 1000000,2000000,3000000,4000000 -t 3 -a ms,mi,hs,qd,rs,cs -g media
```

Además de los códigos fuentes del laboratorio, debe entregar todas las imágenes de las pruebas generadas por `libPlotRuntime`. También debe entregar un archivo llamado `plataforma.txt` que contiene los siguientes datos de la plataforma donde fueron ejecutadas las pruebas:

- Sistema de operación.
- Modelo de CPU.
- Cantidad de memoria RAM del computador.
- Versión del compilador Kotlin.
- Versión de la JVM.

3. Condiciones de entrega

La entrega debe contener los archivos `Ordenamiento.kt`, `PruebaOrdenamiento.kt`, `Makefile`, `probarAlOrd.sh` y `plataforma.txt`, y la carpeta `libPlotRuntime`. También debe entregar las imágenes de las pruebas realizadas, y la declaración de autenticidad firmada. Todo debe estar contenido en un archivo comprimido, con formato `tar.xz`, llamado `LabSem5-X.Y.tar.xz`, donde `X` y `Y` son el número de carné de los estudiante. La entrega debe hacerse por la plataforma Classroom, antes de las 11:50 del día domingo 18 de febrero de 2024.

Referencias

- [1] CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., AND STEIN, C. *Introduction to algorithms*, 4ta ed. MIT press, 2022.