

# Examen 1

## 1. Introducción

El objetivo de este laboratorio es el de implementar una versión modificada de los algoritmos de ordenamiento Mergesort e Insertion sort, tal que usen una nueva relación de orden.

## 2. Actividad a realizar

En primer lugar definimos la nueva relación de orden a utilizar.

**Definición 2.1** (Relación de orden  $\sqsubseteq$ ). *Se define la relación de orden  $\sqsubseteq$  sobre  $\mathbb{Z} \times \mathbb{Z}$  como sigue:*

$$(a, b) \sqsubseteq (a', b') \iff a < a' \vee (a = a' \wedge b \leq b')$$

La relación de orden  $\sqsubseteq$  nos permite ordenar secuencias de elementos que son representados con dos características. Por ejemplo, es posible ordenar una secuencia de personas que están representadas con su edad y peso. Ilustramos el punto con el siguiente ejemplo. Dadas las secuencias **Edad** y **Peso**, se tiene que la posición  $i$  de cada secuencia corresponde al par  $(edad_i, peso_i)$  de la persona  $i$ . Suponga que se tienen las secuencias  $Edad = [20, 19, 23, 19]$  y  $Peso = [60, 71, 58, 65]$ , entonces a la primera persona le corresponde el par  $(20, 60)$ , a la segunda el par  $(19, 71)$ , a la tercera  $(23, 58)$  y a la cuarta  $(19, 65)$ . En este caso se tiene que usando la relación de orden  $\sqsubseteq$ , es posible comparar dos personas a través de su edad y de su peso. Por ejemplo, si ordenamos a las personas caracterizadas con las secuencias **Edad** y **Peso**, usando la relación de orden  $\sqsubseteq$ , entonces tendríamos como resultado las secuencias  $Edad = [19, 19, 20, 23]$  y  $Peso = [65, 71, 60, 58]$ . La Figura 1 muestra la especificación del algoritmo de ordenamiento modificado, usando la relación de orden  $\sqsubseteq$ .

```

[[
  Const  $N$  : Entero;
  Var  $A[1 \dots N], B[1 \dots N]$  : Arreglos de números enteros;
  {Pre:  $N > 0 \wedge MC = \{(A[i], B[i]) \mid 1 \leq i \leq N\}$ }
  algoritmo_ordenamiento_modificado
  {Post:  $\forall_i \mid 2 \leq i \leq N : (A[i-1], B[i-1]) \sqsubseteq (A[i], B[i]) \wedge MC = \{(A[i], B[i]) \mid 1 \leq i \leq N\}$ }
]]

```

Figura 1: Especificación del algoritmo de ordenamiento modificado

Donde  $MC = \{(A[i], B[i]) \mid 1 \leq i \leq N\}$  es un multiconjunto de pares. Para esta actividad algoritmo\_ordenamiento\_modificado corresponde a los algoritmos de ordenamiento a modificar, que son Mergesort e Insertion sort.

Se quiere realice que una librería llamada `order_relation.py`, que contenga las versiones modificadas de Mergesort e Insertion sort, para que ordenen dos secuencias de entrada usando la relación de orden  $\sqsubseteq$ . En específico debe modificar los algoritmos `Insertion-Sort`, de la página 18 de [2], y el `Mergesort-Insertion`, presentado en la página 229 de [1]. Cuando para el algoritmo `Mergesort-Insertion`, el número de elementos a ordenar es menor o igual a 100, entonces se ordena el arreglo con `Insertion-Sort`.

Como segundo ejemplo tenemos que dados los arreglos  $A = [1, 10, 14, 7, 7, 11]$  y  $B = [1, 1, 14, 15, 5, 16]$ , si los arreglos son ordenados por un `algoritmo_ordenamiento_modificado` que sigue la especificación de la Figura 1, entonces se obtienen los arreglos  $A = [1, 7, 7, 10, 11, 14]$  y  $B = [1, 5, 15, 1, 16, 14]$

### 3. Condiciones de entrega

La versión final del código del examen debe estar contenido en un archivo comprimido, con formato *tar.xz*, llamado *Examen1\_X.tar.xz*, donde *X* es el número de carné del estudiante. La entrega del archivo *Examen1\_X.tar.xz*, debe hacerse al profesor del laboratorio, antes de las 1:50 pm del día viernes 14 de febrero de 2020.

### Referencias

- [1] BRASSARD, G., AND BRATLEY, P. *Fundamentals of Algorithmics*. Prentice Hall, 1996.
- [2] CORMEN, T., LEISERSON, C., RIVEST, R., AND STEIN, C. *Introduction to algorithms*, 3rd ed. MIT press, 2009.