

ANÁLISIS DEL RETO

1. Raul Insuasty, <r.insuasty@uniandes.edu.co> , 202015512.

2. Andres Maichel <a.maichel@uniandes.edu.co> , 202225961

3. Andres Jurado <a.jurado@uniandes.edu.co> , 202210996

Requerimiento 1

Descripción

Este requerimiento recibía un año y el código se un sector económico y un año y encuentra la actividad económica con mayor saldo a pagar. Para esto se crea un map cuyas llaves son una concatenación “año-sector económico”, se extraen las actividades económicas y se observa cual es la de mayor saldo a pagar.

Entrada	<i>Mapa, año, sector económico.</i>
Salidas	El elemento con mayor saldo a pagar
Implementado (Sí/No)	Si se implementó por Raul Insuasty.

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Obtener del mapa una lista con las actividades económicas del año y el sector.	$O(1)$
Ordenar la lista de elementos extraída (mergesort) por mayor saldo a pagar.	$O(n \log n)$
Extraer el primer elemento de la lista (get-first)	$O(1)$
TOTAL	$O(n \log n)$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos de entrada fueron el ID 1.

Procesadores	AMD Ryzen 7 4800HS with Radeon Graphics
Memoria RAM	16 GB
Sistema Operativo	Windows 10

Entrada	Tiempo (ms)
small	0.08
5 pct	0.09

10 pct	0.11
20 pct	0.29
30 pct	0.40
50 pct	1.02
80 pct	1.52
large	1.93

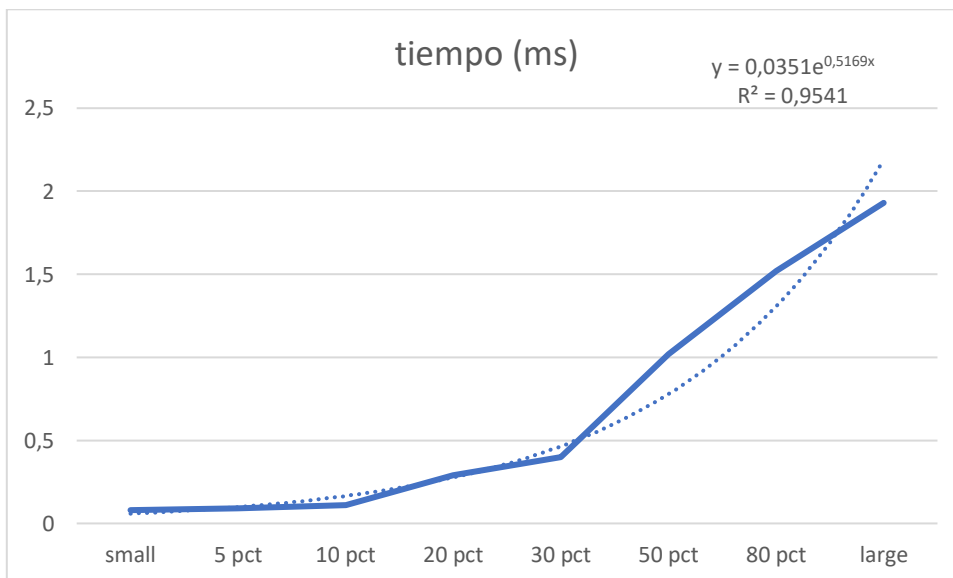
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	0.08
5 pct	Dato2	0.09
10 pct	Dato3	0.11
20 pct	Dato4	0.29
30 pct	Dato5	0.40
50 pct	Dato6	1.02
80 pct	Dato7	1.52
large	Dato8	1.93

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

Como podemos observar el gráfico si se adapta muy bien a la curva por lo que la complejidad es del orden de $O(n \log n)$ dada por el algoritmo de ordenamiento mergesort. El ajuste de la línea parece ser muy cercano dado a que R^2 se acerca mucho a 1.

Los otros elementos del algoritmo parecen no sumar a su ejecución puesto que extraer el elemento del map tiene una complejidad de $O(1)$.

Requerimiento 3

Descripción

Para este requerimiento se utiliza una estructura de datos compuesta; un mapa de años almacena en cada una de sus llaves un mapa de subsectores económicos. El algoritmo entra y extrae los datos de un año y luego toma cada una de las llaves de subsector económico (mapa que esta adentro) y compara sus elementos para determinar cual es el subsector económico el total de retenciones mas bajo..

Entrada	<i>Mapa, año.</i>
Salidas	El elemento con menor total de retenciones.
Implementado (Sí/No)	Si se implementó por Raul Insuasty.

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Extraer el valor del año en el mapa.	$O(1)$
Encontrar el subsector con menor total de retenciones por medio de una iteración en todas las llaves a través de la función <code>find_subsector()</code> . Esta función toma cada llave y suma las retenciones de sus elementos para después comparar cual fue el menor.	$O(n)$
Extraer la lista de elementos con el subsector encontrado del mapa cuyas llaves son los subsectores.	$O(1)$
Ordenar con Mergesort la lista de elementos para sacar las actividades que más y menos aportaron.	$O(n \log n)$
TOTAL	$O(n \log n)$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos de entrada fueron el ID 1.

Procesadores	AMD Ryzen 7 4800HS with Radeon Graphics
Memoria RAM	16 GB

Sistema Operativo	Windows 10
--------------------------	------------

Entrada	Tiempo (ms)
small	0.33
5 pct	0.41
10 pct	0.69
20 pct	0.71
30 pct	0.66
50 pct	0.82
80 pct	1.40
large	1.51

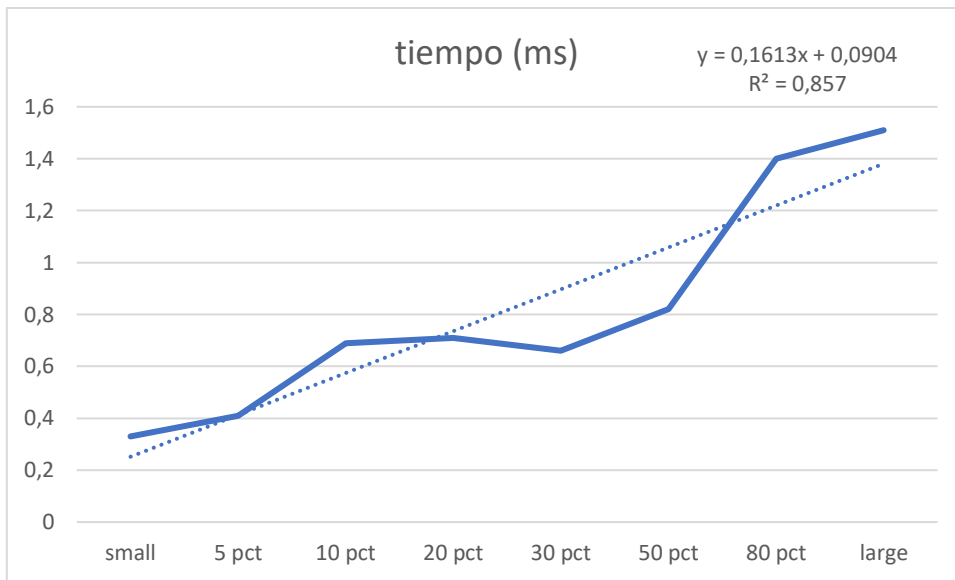
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	0.33
5 pct	Dato2	0.41
10 pct	Dato3	0.69
20 pct	Dato4	0.71
30 pct	Dato5	0.66
50 pct	Dato6	0.82
80 pct	Dato7	1.40
large	Dato8	1.51

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

En este caso el gráfico parece mostrar una tendencia lineal, esto puede deberse a que cuando se calculó la complejidad, el mergesort (cuya complejidad es $O(n \log n)$) ordenó una lista muy pequeña de datos. Gracias a esto puede que no afecte tanto en la complejidad y finalmente acabe siendo de $O(n)$.