

ANÁLISIS DEL RETO

Ana Sofía Orejuela, 202215562, a.orejuelar@uniandes.edu.co

Juan Camilo López, 202122845, jc.lopezc1@uniandes.edu.co

Nicolás Bedoya 202122845, n.bedoyaf@uniandes.edu.co

Requerimiento 1

Descripción

Se filtra el mapa construido a partir del “release_dates” teniendo en cuenta sólo los registros en el rango ingresado. Se crea una lista y un contador, mientras el registro no esté vacío, se añade a la nueva lista y se suman los registros al contador. Por último, la lista se ordena con un merge sort por fecha.

Entrada	<ul style="list-style-type: none">• Plataforma (Platforms)• Límite inferior de fecha de lanzamiento (Release_Date).• Límite superior de fecha de lanzamiento (Release_Date).
Salidas	<ul style="list-style-type: none">• El número total de videojuegos disponibles en la plataforma.• El número de videojuegos disponibles en el rango de fechas de publicación para la plataforma.• Los 3 primeros y últimos registros disponibles en dicho rango. Cada uno de los videojuegos deberá mostrar la siguiente información:<ul style="list-style-type: none">o Nombre del videojuego.o Abreviación.o Género del videojuegoo Plataformas disponibles.o Cantidad total de intentos para romper un récord en el videojuego.o Fecha de publicación.

Implementado (Sí/No)	Si, Grupo
----------------------	-----------

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Fun1	$O(N^2)$
TOTAL	$O(N^2)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procedimiento: aplicar `start_time = getTime()` al principio de las funciones utilizadas en model y antes de retornar, utilizar `end_time = getTime()` y `delta_Time = deltaTime(start_time,end_time)`

Condiciones: entre 2000-01-01 y el 2005-12-31

Entrada/cantidad de archivos cargados	Tiempo (ms)
0.5% = 229	0.75
5% = 1778	2.55
10% = 3248	3.29
20% = 5938	3.97
30% = 8537	4.24
50% = 13651	4.27
80%=21265	4.10
100%=26333	4.16

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Fun1

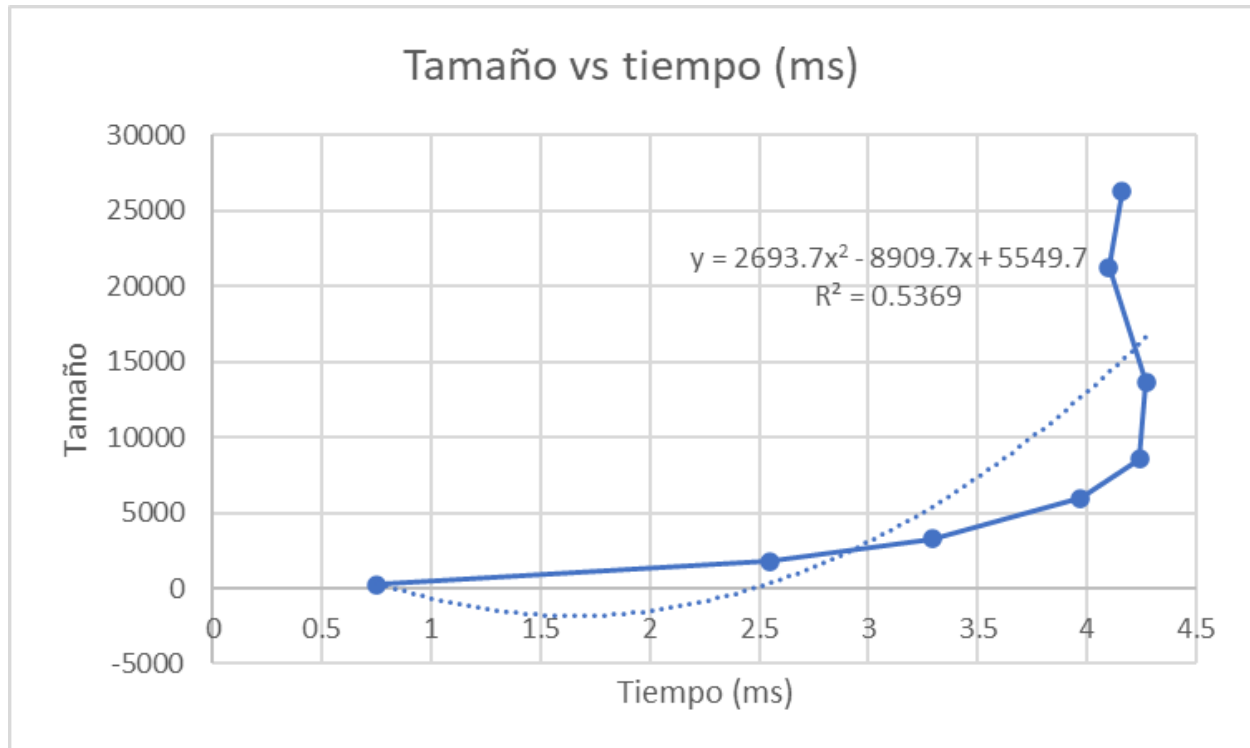
Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	0.75
5% = 1778	2.55
10% = 3248	3.29
20% = 5938	3.97
30% = 8537	4.24
50% = 13651	4.27

80%=21265	4.10
100%=26333	4.16

Graficas

Las gráficas con la representación de las pruebas realizadas.

Fun1



Análisis

Este requerimiento posee una complejidad temporal de N^2 debido a que se implementa un doble ciclo que recorre los registros en las fechas seleccionadas.

Requerimiento 2

Descripción

Se recorre el mapa de tiempos, luego si el jugador coincide con el enviado por parámetro se suma uno al contador y se añaden los registros del jugador a una nueva lista. Posteriormente esta lista se ordena con un merge sort según el tiempo.

Entrada	Nombre del jugador (Player_0).
----------------	--------------------------------

Salidas	<p>El número de registros del jugador en donde obtuvo el mejor tiempo (Time_0).</p> <ul style="list-style-type: none"> • El número de intentos (Num_Runs) que ha realizado el jugador para obtener el mejor tiempo. • Los 5 registros con el menor tiempo (Time_0) registrado por el jugador. Cada uno de los registros deberá mostrar la siguiente información: <ul style="list-style-type: none"> o Nombre del videojuego. o Nombre de la categoría. o Nombre de la subcategoría. o Número de intentos de romper el récord. o Nombre del jugador o Nacionalidad del jugador o Tiempo obtenido y fecha en que obtuvo del récord.
Implementado (Sí/No)	Si, Grupo

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Fun1	$O(N)$
TOTAL	$O(N)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procedimiento: aplicar start_time = getTime() al principio de las funciones utilizadas en model y antes de retornar, utilizar end_time = getTime() y delta_Time = deltaTime(start_time,end_time)

Condiciones: Flamming

Entrada/Tamaño archivos cargados	Tiempo (ms)
---	--------------------

0.5% = 229	68.14
5% = 1778	601.46
10% = 3248	1135.34
20% = 5938	2261.73
30% = 8537	2991.22
50% = 13651	4972.33
80%=21265	7133
100%=26333	8837.84

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

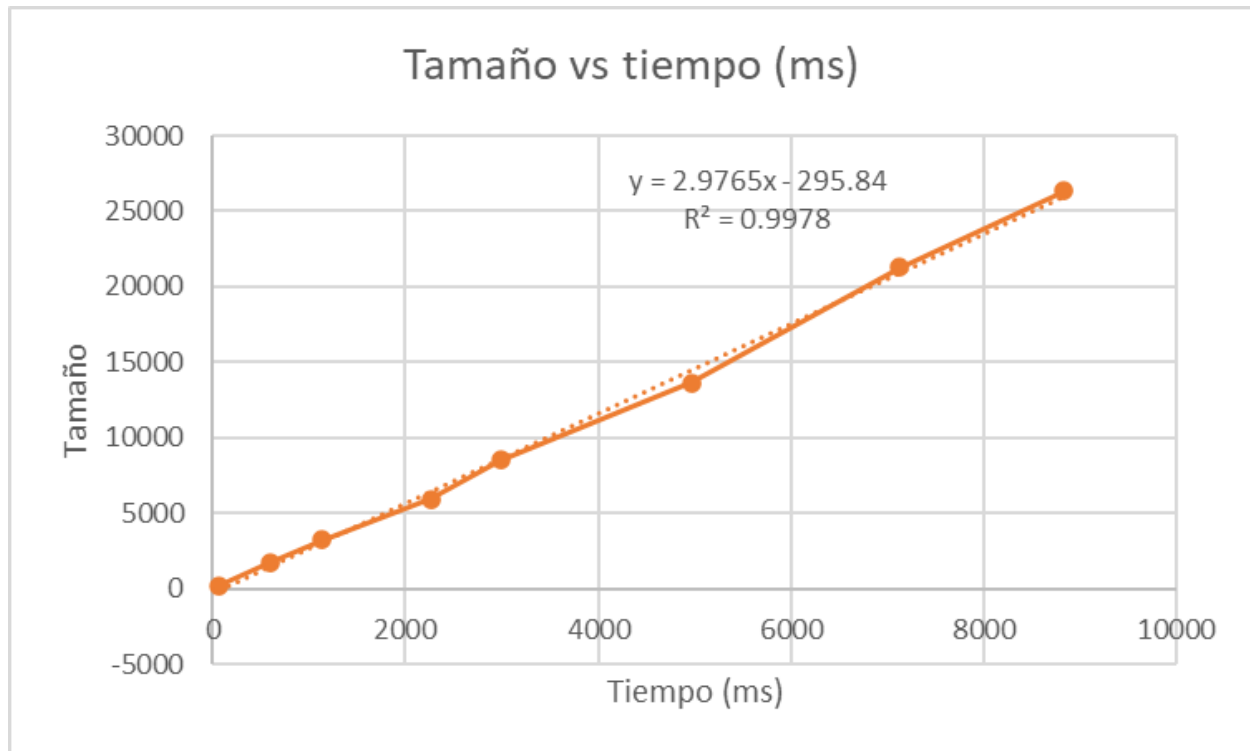
Fun1

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	68.14
5% = 1778	601.46
10% = 3248	1135.34
20% = 5938	2261.73
30% =8537	2991.22
50% = 13651	4972.33
80%=21265	7133
100%=26333	8837.84

Gráficas

Las gráficas con la representación de las pruebas realizadas.

Fun1



Análisis

Justo como se mencionó anteriormente esta función tiene crecimiento lineal. Ahora bien, esto se debe a que, a pesar de haber un triple ciclo, el primero se ejecuta N veces, los demás una fracción mucho más pequeña de N. Esto hace que la función sea de carácter lineal.

Requerimiento 3

Descripción

Se recorre el mapa de num_runs que tiene valores entre los dados por parámetro. Se busca el menor y se añade el registro a una lista.

Entrada	<ul style="list-style-type: none">• Límite inferior del número de intentos para romper el récord (Num_Runs).• Límite superior del número de intentos para romper el récord (Num_Runs).
Salidas	<ul style="list-style-type: none">• El número de registros que cumplen con los criterios del rango de búsqueda.• Los 3 primeros y últimos registros disponibles en dicho rango. Cada uno de los registros deberá mostrar la siguiente información:<ul style="list-style-type: none">o Nombre del videojuego.

	o Nombre de la categoría. o Nombre de la subcategoría. o Número de intentos de romper el récord. o Nombre del jugador o Nacionalidad del jugador o Valor del mejor tiempo obtenido y fecha en que obtuvo del récord.
Implementado (Sí/No)	Si, Ana Sofía

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Fun1	$O(N \log N)$
TOTAL	$O(N \log N)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procedimiento: aplicar `start_time = getTime()` al principio de las funciones utilizadas en model y antes de retornar, utilizar `end_time = getTime()` y `delta_Time = deltaTime(start_time,end_time)`

Condiciones: 21-75

Procesadores	AMD Ryzen 9 5900X 12-Core Processor 3.70 GHz
Memoria RAM (GB)	32.0 GB
Sistema Operativo	Windows 64bit

Entrada/Tamaño archivos cargados	Tiempo (ms)(suma tiempo cada función)
0.5% = 229	0.15

5% = 1778	0.42
10% = 3248	0.51
20% = 5938	0.62
30% = 8537	0.63
50%13651	0.69
80%=21265	0.65
100%=26333	0.763

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

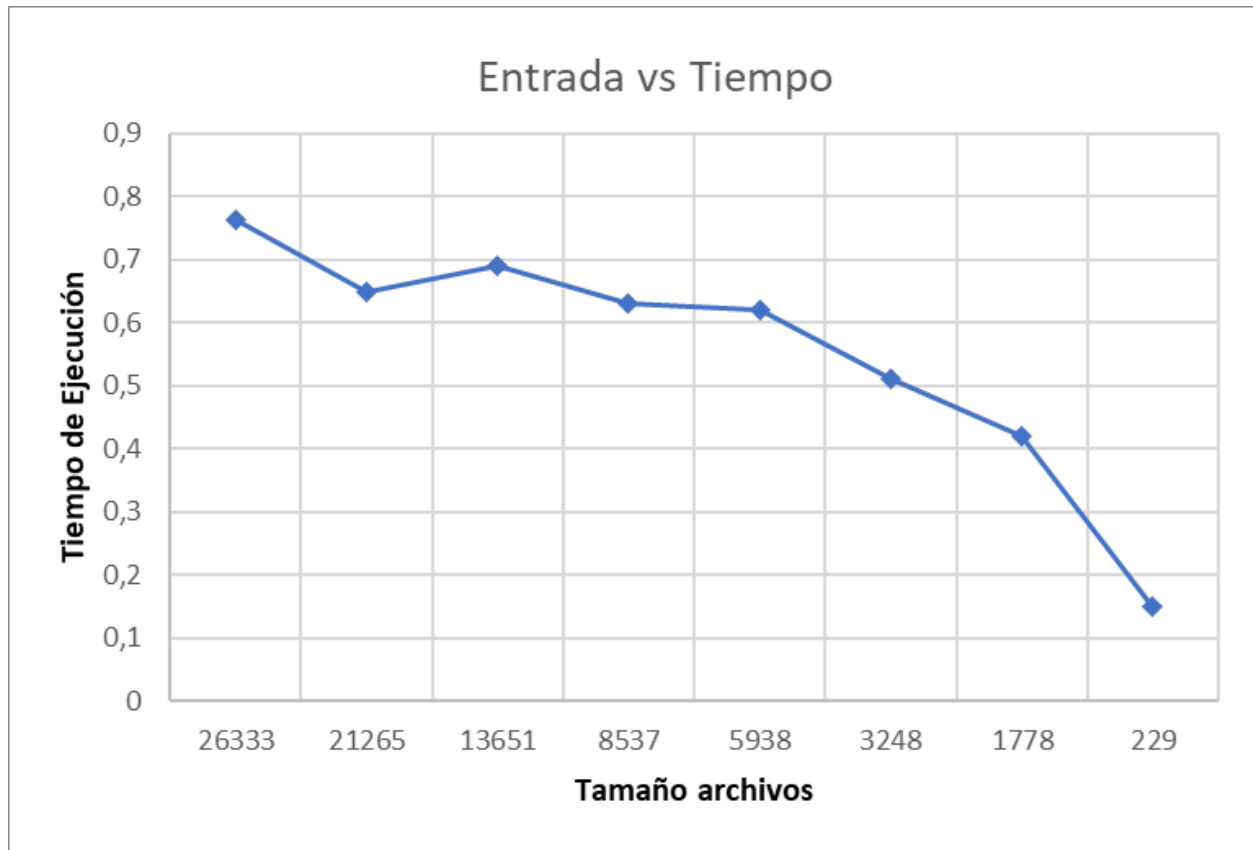
Fun1

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% =229	0.15
5% = 1778	0.42
10% = 3248	0.51
20% = 5938	0.62
30% = 8537	0.63
50%13651	0.69
80%=21265	0.65
100%=26333	0.763

Gráficas

Las gráficas con la representación de las pruebas realizadas.

Fun1



Análisis

En este requerimiento se puede ver como la gráfica tiende a tener una forma del tipo $(O)\log N$, sin embargo, esto se puede deber principalmente a que el rango que se utilizó para esta prueba es relativamente pequeño. De todas maneras, el requerimiento posee una complejidad temporal de $(O)N\log N$ ya que mientras se recorren todos los registro (N en el peor caso) se toma el $\minkey(\log N)$.

Requerimiento 4

Descripción

Se recorre el mapa de num_runs que tiene valores entre los dados por parámetro. Se busca el máximo y se añade el registro a una lista.

Entrada	<ul style="list-style-type: none">• Límite inferior de la fecha: hora en que se obtuvo el récord (Record_Date_0).• Límite superior de la fecha: hora de que se obtuvo el récord (Record_Date_0).
---------	---

Salidas	<ul style="list-style-type: none"> • El número de registros que cumplen con los criterios del rango de búsqueda. • Los 3 primeros y últimos registros disponibles en dicho rango. Cada uno de los registros deberá mostrar la siguiente información: <ul style="list-style-type: none"> o Nombre del videojuego. o Nombre de la categoría. o Nombre de la subcategoría. o Número de intentos de romper el récord. o Nombre del jugador o Nacionalidad del jugador o Valor del mejor tiempo obtenido y fecha en que obtuvo del récord.
Implementado (Sí/No)	Si, Juan Camilo

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Fun1	$O(N \log N)$
TOTAL	$O(N \log N)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procedimiento: aplicar `start_time = getTime()` al principio de las funciones utilizadas en model y antes de retornar, utilizar `end_time = getTime()` y `delta_Time = deltaTime(start_time, end_time)`

Condiciones: 2019-03-06T04:03:53Z, 2021-10-17T15:48:00Z

Procesadores	AMD Ryzen 9 5900X 12-Core Processor 3.70 GHz
Memoria RAM (GB)	32.0 GB

Sistema Operativo	Windows 64bit
--------------------------	---------------

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	0.51
5% = 1778	4.11
10% = 3248	8.54
20% = 5938	14.55
30% = 8537	22.72
50% = 13651	38.26
80%=21265	56.16
100%=26333	73.75

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

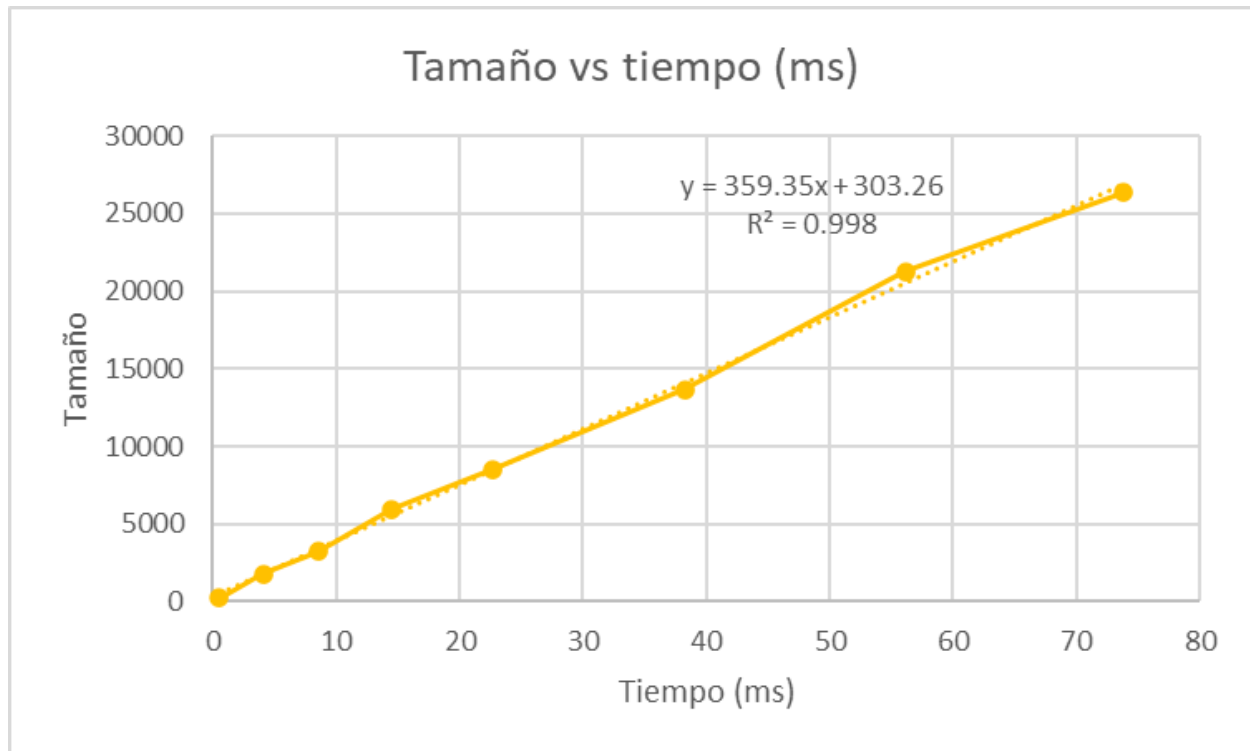
Fun1

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	0.51
5% = 1778	4.11
10% = 3248	8.54
20% = 5938	14.55
30% = 8537	22.72
50% = 13651	38.26
80%=21265	56.16
100% =26333	73.75

Graficas

Las gráficas con la representación de las pruebas realizadas.

Fun1



Análisis

En este requerimiento se puede ver como la gráfica tiende a tener una forma del tipo $(O)\log N$, sin embargo, esto se puede deber principalmente a que el rango que se utilizó para esta prueba es relativamente pequeño. De todas maneras, el requerimiento posee una complejidad temporal de $(O)N\log N$ ya que mientras se recorren todos los registro (N en el peor caso) se toma el $\maxkey(\log N)$.

Requerimiento 5

Descripción

Se recorre el mapa de times_0 que tiene valores entre los dados por parámetro. Se busca el máximo y se añade el registro a una lista.

Entrada	<ul style="list-style-type: none"> • Límite inferior de la duración para el mejor tiempo registrado (Time_0). • Límite superior de la duración para el mejor tiempo registrado (Time_0).
----------------	--

Salidas	<ul style="list-style-type: none"> • El número de registros que cumplen con los criterios del rango de búsqueda. • Los 3 primeros y últimos registros disponibles en dicho rango. Cada uno de los registros deberá mostrar la siguiente información: <ul style="list-style-type: none"> o Nombre del videojuego. o Nombre de la categoría. o Nombre de la subcategoría. o Número de intentos de romper el récord. o Nombre del jugador o Nacionalidad del jugador o Valor del mejor tiempo obtenido y fecha en que obtuvo del récord.
Implementado (Sí/No)	Si, Nicolás

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Paso 1	$O(N \log N)$
TOTAL	$O(N \log N)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procedimiento: aplicar `start_time = getTime()` al principio de las funciones utilizadas en model y antes de retornar, utilizar `end_time = getTime()` y `delta_Time = deltaTime(start_time,end_time)`

Condiciones: 542.10, 1887.50

Procesadores	Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz
Memoria RAM (GB)	12.0 GB
Sistema Operativo	Windows 64bit

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	0.58
5% = 1778	1.90

10% = 3248	4.08
20% = 5938	8.20
30% =8537	9.63
50%=13651	18.14
80%=21265	20.00
100%=26333	27.97

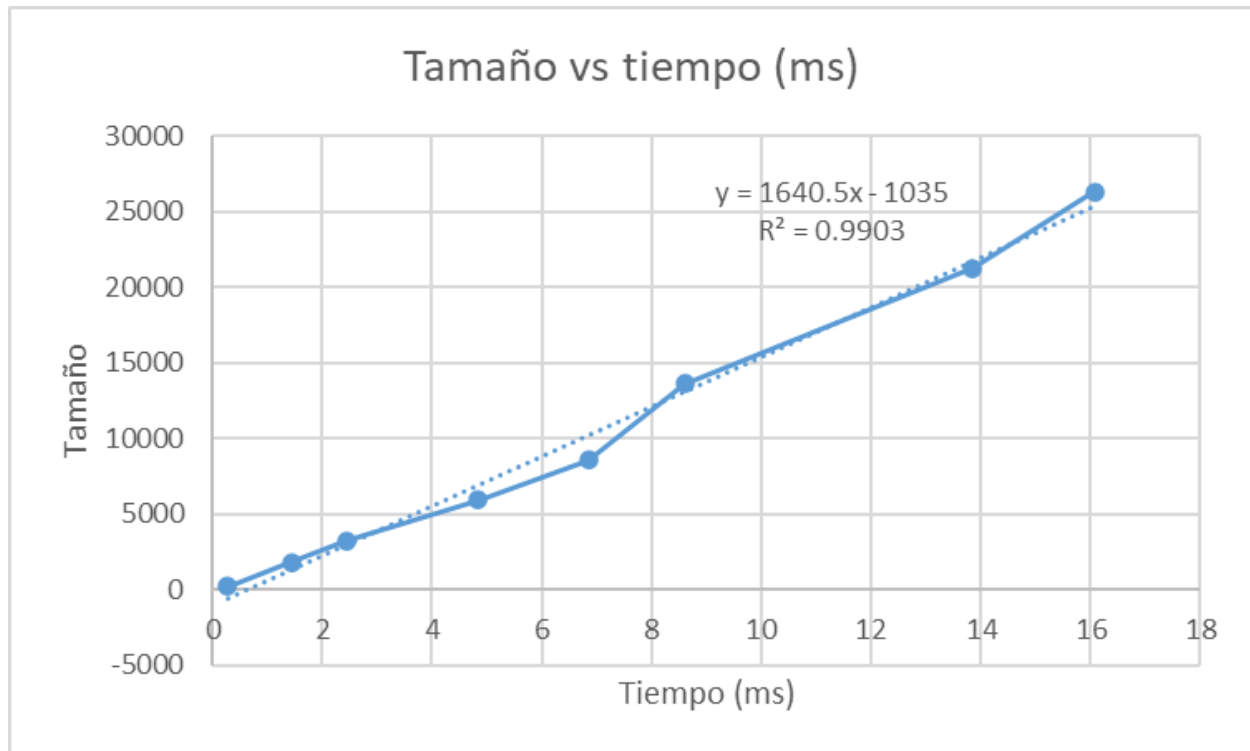
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	0.58
5% = 1778	1.90
10% = 3248	4.08
20% = 5938	8.20
30% = 8537	9.63
50%=13651	18.14
80%=21265	20.00
100%=26333	27.97

Graficas

Fun1



Análisis

En este requerimiento se puede ver como la gráfica tiende a tener una forma del tipo $(O)\log N$, sin embargo, esto se puede deber principalmente a que el rango que se utilizó para esta prueba es relativamente pequeño. De todas maneras, el requerimiento posee una complejidad temporal de $(O)N\log N$ ya que mientras se recorren todos los registro (N en el peor caso) se toma el $\maxkey(\log N)$.

Requerimiento 6

Descripción

Entrada	<ul style="list-style-type: none"> • Límite inferior del año de lanzamiento (Tomado del Release_Date). • Límite superior del año de lanzamiento (Tomado del Release_Date). • Numero de segmentos en que se divide el rango de propiedad en el histograma (N) • Numero de niveles en que se dividen las marcas de jugadores en el histograma (x). • La opción de consultar las siguientes propiedades de los registros: <ul style="list-style-type: none"> o El mejor tiempo registrado (Time_0), el segundo mejor tiempo (Time_1), el tercer mejor tiempo (Time_2). o El tiempo promedio registrado (el promedio de Time_0, Time_1, y Time_2). o El número de intentos registrados (Num_Runs).
Salidas	<ul style="list-style-type: none"> • El número total de registros consultados. • El número total de registros incluidos en el conteo para graficar el histograma. • Valor mínimo y valor máximo dentro del rango del histograma. • El histograma con la distribución de la propiedad dividido por rango y niveles.
Implementado (Sí/No)	Si, Grupo

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Fun1	$O(N^2 \log N)$
TOTAL	$O(N^2 \log N)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procedimiento: aplicar start_time = getTime() al principio de las funciones utilizadas en model y antes de retornar, utilizar end_time = getTime() y delta_Time = deltaTime(start_time,end_time)

Condiciones: 2017-2021, 5 segmentos, 7 niveles, Time_avg

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	1.34
5% = 1778	11.07
10% = 3248	21.12
20% = 5938	42.63
30% = 8537	65.72
50%=13651	112.10
80%=21265	183.08
100%=26333	236.40

Tablas de datos

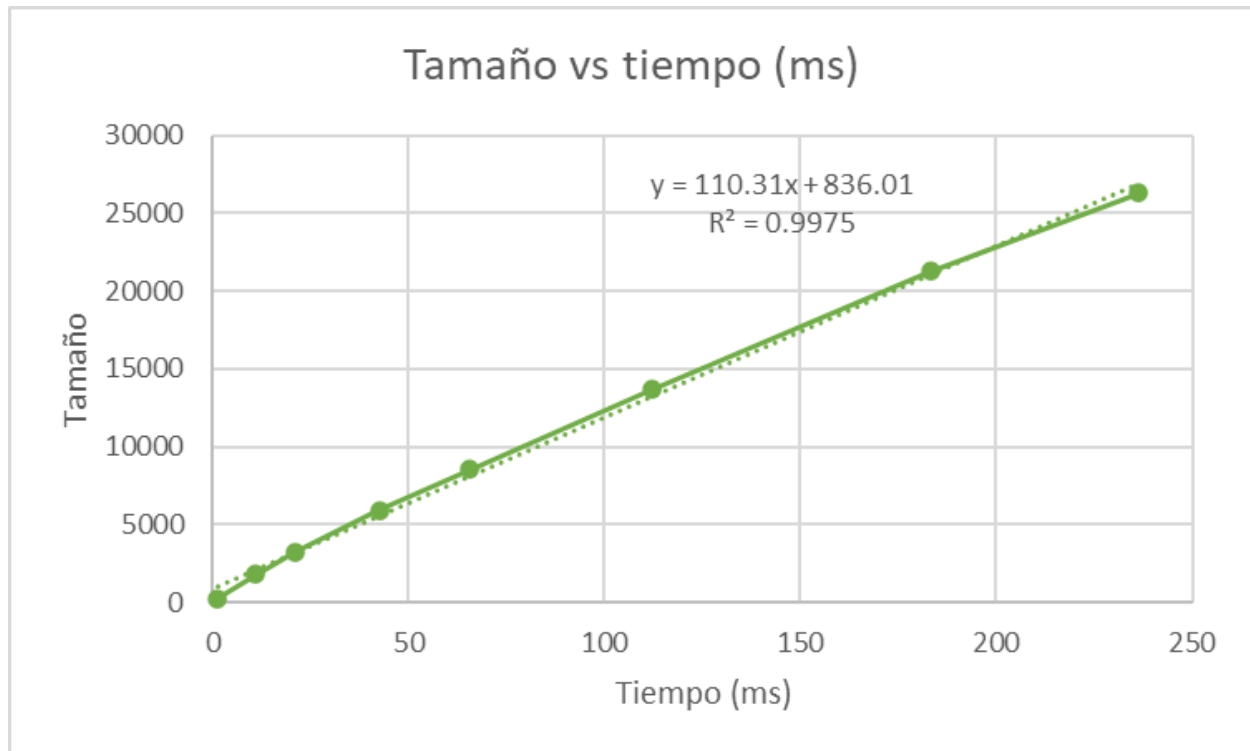
Las tablas con la recopilación de datos de las pruebas.

Fun1

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	1.34
5% = 1778	11.07
10% = 3248	21.12
20% = 5938	42.63
30% = 8537	65.72
50%=13651	112.10
80%=21265	183.08
100%=26333	236.40

Gráficas

Fun1



Análisis

Nuevamente, al ser N un número tan pequeño gracias al rango seleccionado para las pruebas, se logra observar una función que tiende a $(O)\log N$. Sin embargo, el requerimiento posee una complejidad temporal de $O(N^2 \log N)$ ya que mientras se recorren todos los registros y sus valores se usa put o get $((O)\log N)$

Requerimiento 7

Descripción

Entrada	<ul style="list-style-type: none"> • Plataforma de interés (Platforms). • TOP N de los videojuegos para retransmitir (N)
Salidas	<ul style="list-style-type: none"> • El número total de videojuegos disponibles en dicha plataforma. • Los N videojuegos más rentables para transmitir. Cada uno de los videojuegos debe tener la siguiente información: <ul style="list-style-type: none"> o Nombre del videojuego. o Abreviación. o Género del videojuego

	o Cantidad total de intentos para romper un récord en el videojuego. o Fecha de publicación. o Nombres de los jugadores con mejor tiempo. o Dinero esperado por la transmisión del videojuego
Implementado (Sí/No)	Si, Grupo

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Fun1	$O(N^3)$
TOTAL	$O(N^3)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procedimiento: aplicar `start_time = getTime()` al principio de las funciones utilizadas en model y antes de retornar, utilizar `end_time = getTime()` y `delta_Time = deltaTime(start_time,end_time)`

Condiciones: PC, top 5

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	3.81
5% = 1778	95.86
10% = 3248	256.57
20% = 5938	611.17
30% = 8537	908.29
50%=13651	1654.57
80%=21265	2586.86
100%=26333	3226.18

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

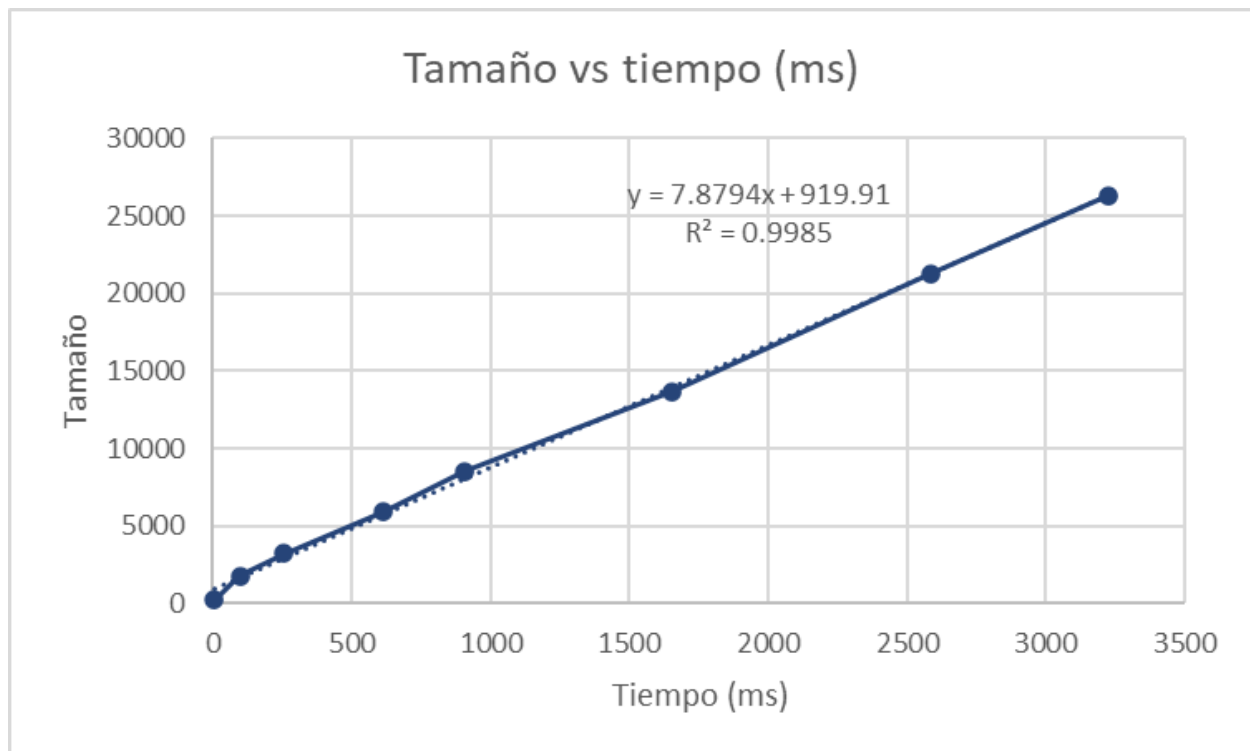
Fun1

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	3.81
5% = 1778	95.86
10% = 3248	256.57
20% = 5938	611.17
30% = 8537	908.29
50%=13651	1654.57
80%=21265	2586.86
100%=26333	3226.18

Graficas

Las gráficas con la representación de las pruebas realizadas.

Fun1



Análisis

La complejidad de este requerimiento es de N^3 debido a que se recorren ambos mapas al tiempo y se usa un merge sort que añade otra N a la complejidad.

Requerimiento 8

Descripción

Breve descripción de cómo abordaron la implementación del requerimiento

Entrada	<ul style="list-style-type: none">• Año de publicación sobre el cual se quiere obtener los histogramas (Tomado del Release_Date)• Límite inferior de la duración del mejor tiempo del récord (Time_0).• Límite superior de la duración del mejor tiempo del récord (Time_0).
Salidas	<ul style="list-style-type: none">• El número total de registros de speedrun en dicho año y rango.• Un mapa interactivo de clústeres que muestre todos los registros de speedrun en dicho año y rango.
Implementado (Sí/No)	Si, Grupo

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Fun1	$O(N^3)$
TOTAL	$O(N^3)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procedimiento: aplicar `start_time = getTime()` al principio de las funciones utilizadas en model y antes de retornar, utilizar `end_time = getTime()` y `delta_Time = deltaTime(start_time,end_time)`

Condiciones: 2017, 1000-5000

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	3.6347
5% = 1778	8125.01
10% = 3248	12250.15
20% = 5938	16599.68
30% = 8537	17551.11

50%=13651	21598.61
80%=21265	25101.26
100%==26333	26575.50

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

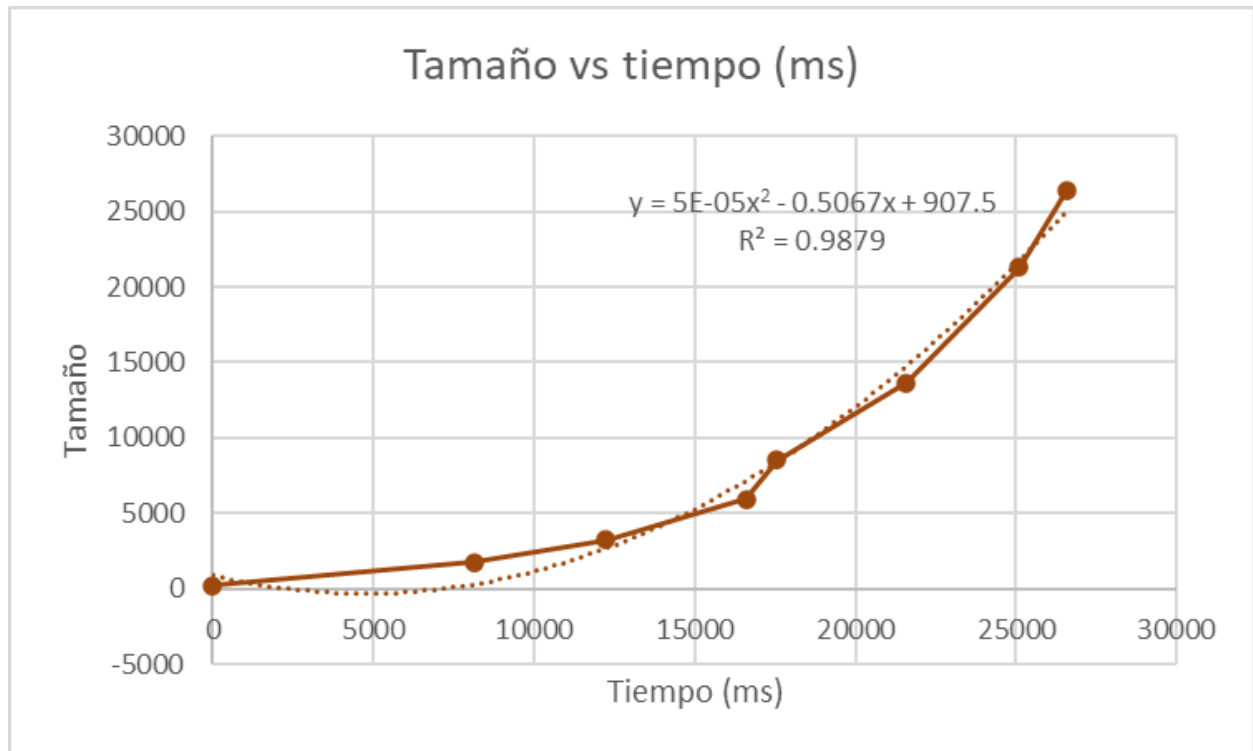
Fun1

Entrada/Tamaño archivos cargados	Tiempo (ms)
0.5% = 229	0.883
5% = 1778	8125.01
10% = 3248	12250.15
20% = 5938	16599.68
30% = 8537	17551.11
50%=13651	21598.61
80%=21265	25101.26
100%=26333	26575.50

Gráficas

Las gráficas con la representación de las pruebas realizadas.

Fun1



Análisis

La complejidad temporal de esta gráfica termina por ser $(N^3 \log N)$ debido a que en el código se usa un triple ciclo y según la condición que se cumpla se usa un $\text{put}(\log N)$.