**ANÁLISIS DEL RETO**

Andrés Ignacio Tello umerez, 202223807, [a.tellou@uniandes.edu.co](mailto:a.tellou@uniandes.edu.co)

*Emile Bottagisio, 202223504*, e.bottagisio@uniandes.edu.co

Juan Esteban Guzmán, *202313334*, je.guzman2@uniandes.edu.co

# **Carga de datos**

## **Descripción**

Se cargan los datos del csv’s de los earthqueakers.A screen shot of a computer program

Description automatically generated

|  |  |
| --- | --- |
| **Entrada** | El archivo csv |
| **Salidas** | El total de los registros cargados desde el archivo. Mostrar los primeros 5 y últimos 5 registros de eventos sísmicos cargados con las siguientes características:  o La fecha y hora del evento (time).  o La Latitud donde ocurrió el evento (lat).  o La longitud donde ocurrió el evento (long).  o La profundidad del evento (depth).  o La magnitud del evento (mag).  o La significancia del evento (sig).  o El número de estaciones que registraron el evento (nst).  o La distancia azimutal del evento (gap).  o El título del evento sísmico (title).  o El número de reportes del sistema DYFI asociados al evento (felt). o La intensidad máxima del evento reportada por el sistema DYFI (cdi).  o La Intensidad máxima instrumental estimada para el evento (mmi).  o Marca si el evento ocurrió en una región oceánica (tsunami). |
| **Implementado (Sí/No)** | Si |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 (Operaciones sobre listas de temblores): La operación lt.addLast(lista, data), tiene una complejidad O(n) en el peor de los casos, donde n es el número de elementos en la lista | O(N) |
| Paso 2 (Operaciones sobre el mapa de magnitudes): - - - Esta tiene 2 operaciones om.contains(mapa\_magnitud, magnitud) y om.get(mapa\_magnitud, magnitud) tiene una complejidad O(1) y en el peor caso es O(N).  - La operación lt.addLast(magnitud\_search, data) tiene una complejidad O(1) y depende de la implementación de la lista | O(N) |
| Paso 3:  -Las operaciones om.contains(mapa\_profundidad, profundidad) y om.get(mapa\_profundidad, profundidad) tienen una complejidad similar a las operaciones del mapa de magnitudes.  -La operación lt.addLast(profundidad\_search, data) tiene una complejidad  O(1) en promedio, dependiendo de la implementación de la lista. | O(1) |
| ***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).

|  |  |
| --- | --- |
| Procesadores | AMD Ryzen 7 5700HS with Radeon Graphics(16CPUS) -3.8Ghz |
| Memoria RAM | 8 GB |
| Sistema Operativo | Windows 11 |

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (ms)** |
| small | 146.394 |
| 5pct | 689.38 |
| 10pct | 1390.478 |
| 20pct | 2247.901 |
| 30pct | 2534.797 |
| 50pct | 6056.217 |
| 80pct | 10079.920 |
| large | 11763.434 |

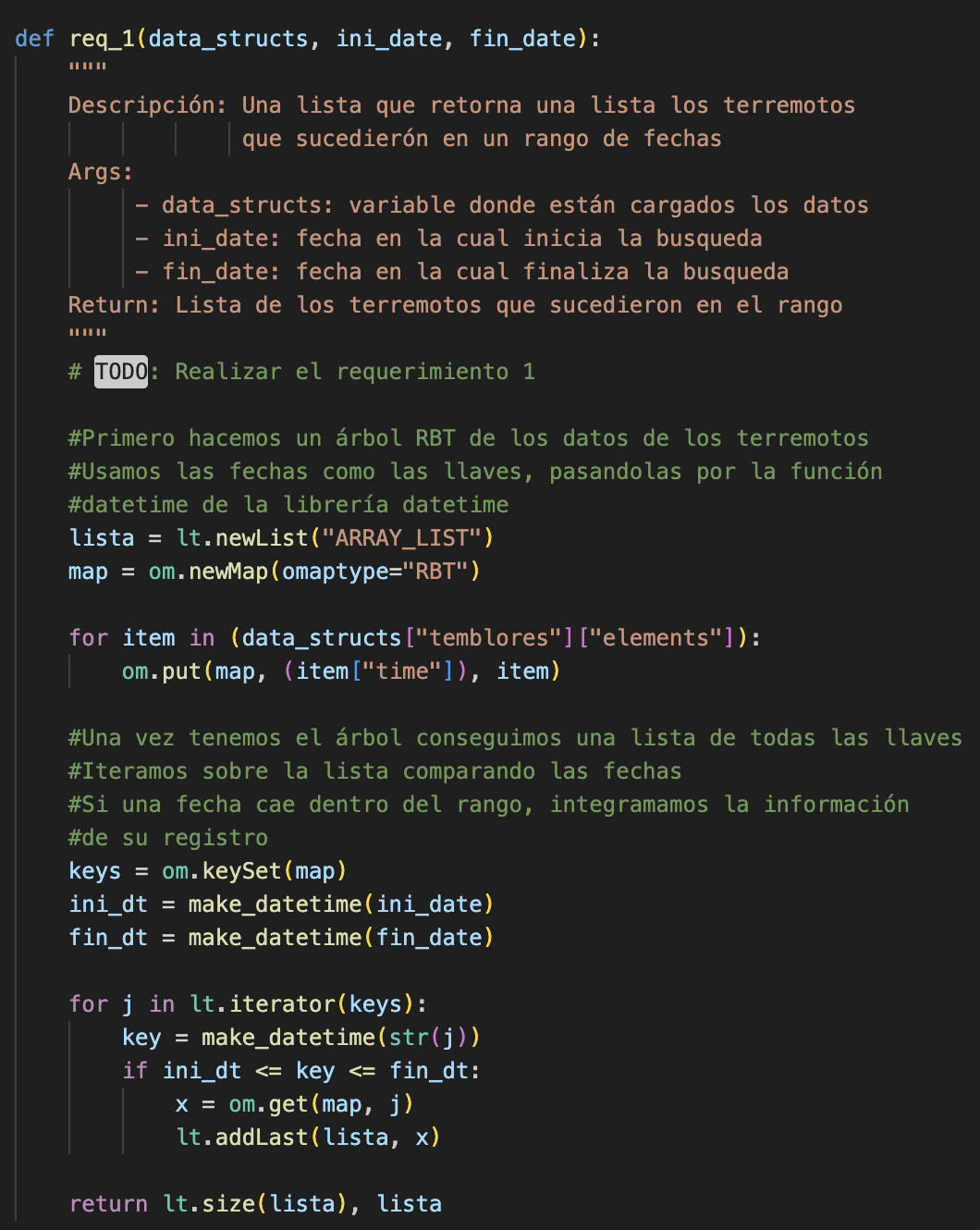
### **Gráficas**

A graph with a line

Description automatically generated

# **Requerimiento <<1>>**

## **Descripción**



Este requerimiento se encarga de retornar una lista de terremotos en un rango de fechas. Lo primero que hacemos es hacer un arbol RBT y una lista, la lista sera lo que retornaremos y el arbol es nuestra estructura de datos. En la función sacamos una lista de las llaves del arbol que son las fechas de los terremotos. Despues recorremos la lista consiguiendo los registros que caen entre las dos fechas. Al final retorna la lista con los registros y el temaño de la lista que corresponde a la cantidad de registros.

|  |  |
| --- | --- |
| **Entrada** | * Control * Fecha inicial * Fecha final |
| **Salidas** | * Tamaño de la lista * Lista con datos |
| **Implementado (Sí/No)** | Si se implemento, lo implemento Andrés Tello |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 (Hacer la estructura de datos) | O(n) (n=cantidad de registros) |
| Paso 2 (Iterar por la lista) | O(n) |
| ***TOTAL*** | ***O(n2)*** |

## **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).

|  |  |
| --- | --- |
| Procesadores | Apple M1 |
| Memoria RAM | 16 GB |
| Sistema Operativo | MacOs Monterrey |

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| large | 1.602 |
| 80% | 1.641 |
| 50% | 1.585 |
| 30% | 1.605 |
| 20% | 1.049 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.

A graph with a line and a line

Description automatically generated

## **Análisis**

A pesar de tener una complejidad de O(n2), esta función incrementa muy poco debido a que la iteración para buscar los datos no suele ser extremadamente grande. Dicho eso, en la grafica la duración parece cambiar bastante y de manera erratica. Sin embargo, la regresión lineal muestra que el incremento de la duración con respecto al porcentaje es muy leve.

Tambien es importante considerar la posibilidad de que los parametros de prueba fueran unos que no le exigen más al computador a medida de que cambiamos de archivo. Es posible que estop suceda si la diferencia entre las fechas es muy pequeña o si no incrementa mucho la cantidad de datos que hay

# **Requerimiento 3**

## **Descripción**

A screen shot of a computer code

Description automatically generated

Consultar los 10 eventos sísmicos más recientes que superen una magnitud mínima y no superen una profundidad máxima indicada.

|  |  |
| --- | --- |
| **Entrada** | • La magnitud mínima del evento (mag).  • La profundidad máxima del evento (depth). |
| **Salidas** | La respuesta esperada debe contener:  • El número total de eventos sísmicos registrados dentro de los límites de magnitud y profundidad indicados.  • Los diez (10) eventos cronológicamente más recientes que cumplan con las condiciones de profundidad y magnitud indicados. Cada uno de los eventos en la consulta debe desplegar la siguiente información:  o La fecha y hora del evento (time). o La magnitud del evento (mag) o La latitud donde ocurrió el evento (lat).  o La longitud donde ocurrió el evento (long).  o La profundidad del evento (depth). o La significancia del evento (sig).  o La distancia azimutal del evento (gap).  o El número de estaciones utilizadas para medir el evento (nst).  o El título del evento sísmico (title). o La intensidad máxima del evento reportada por el sistema DYFI (cdi).  o La intensidad máxima instrumental estimada para el evento (mmi).  o El algoritmo de cálculo de magnitud del evento (magType).  o El tipo del evento sísmico (type). o El código del evento (code). |
| **Implementado (Sí/No)** | Si Juan Esteban Guzman |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Iteración a través de analyzer['earthquakes']: La complejidad de esta parte depende del tamaño de analyzer['earthquakes']. Si hay 'n' eventos en esta lista, la complejidad sería O(n). | O(n) |
| Filtrado de eventos: Se filtran los eventos basados en dos condiciones (evento['mag'] >= min\_magnitude y evento['depth'] <= max\_depth). Si 'm' eventos cumplen ambas condiciones, la complejidad de esta parte sería O(m). | O(n) |
| Ordenamiento de la lista eventos\_filtrados: Usando merg.sort para ordenar la lista de eventos filtrados. El costo de esta operación de ordenamiento depende del algoritmo de ordenamiento utilizado. Si se utiliza un algoritmo de ordenamiento eficiente como Merge Sort o Quick Sort, la complejidad sería O(m log m), donde 'm' es la cantidad de eventos filtrados. | O(n log n) |
| Creación de eventos\_resultado: Se realiza una operación lt.subList para obtener una sublista de eventos\_filtrados. Esta operación tiene una complejidad O(k), donde 'k' es la longitud de la sublista (en este caso, 10). | O(n) |
| ***TOTAL*** | ***O(N log N)*** |

## **Pruebas Realizadas**

|  |  |
| --- | --- |
| Procesadores | AMD Ryzen 7 5700HS with Radeon Graphics(16CPUS) -3.8Ghz |
| Memoria RAM | 8 GB |
| Sistema Operativo | Windows 11 |

### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|  |  |  |
| --- | --- | --- |
| **Muestra** | **Salida** | **Tiempo (ms)** |
| small | Dato1 | 143,141 |
| 5 pct | Dato2 | 512,991 |
| 10 pct | Dato3 | 1411,062 |
| 20 pct | Dato4 | 2218,194 |
| 30 pct | Dato5 | 2411,736 |
| 50 pct | Dato6 | 5291,514 |
| 80 pct | Dato7 | 9281,821 |
| large | Dato8 | 11201,934 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.

A graph with a line

Description automatically generated

# **Requerimiento <<4>>**

## **Descripción**

A screen shot of a computer program

Description automatically generated

Esta función se encarga de retornar los terremotos que cumplen con un minimo de significancia y un maximo de distancia azimutal. La funcion primero hace un arbol tipo RBT que usa sig como la llave. Con la estructura de datos, la función consigue la llave mas grande en el arbol y usa eso y los parametros dados por el usuario para conseguir los registros que tienen sig más alto que el parametro y menor que la mayor llave. Al final, hace un MergeSort y reemplaza todos los valores vacios por “Unknown”. Finalmente, se retorna el tamaño de la lista y la lista como tal.

|  |  |
| --- | --- |
| **Entrada** | * Control * La significancia minima * La distancia azimutal maxima |
| **Salidas** | * Tamaño de la lista * Lista con los datos |
| **Implementado (Sí/No)** | Si se implento, lo implemento Andrés Tello |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 (Hacer la estructura) | O(n) |
| Paso 2 (Organizar la lista) | O(Log(n)\*n) |
| Paso 3 (Iterar por la lista de llaves) | O(n) |
| ***TOTAL*** | ***O(Log(n)\*n3)*** |

## **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).

|  |  |
| --- | --- |
| Procesadores | Apple M1 |
| Memoria RAM | 16 GB |
| Sistema Operativo | MacOs Monterrey |

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| large | 0.488 |
| 80% | 0.494 |
| 50% | 0.484 |
| 30% | 0.476 |
| 20% | 0.476 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.

A graph with a line and a line

Description automatically generated with medium confidence

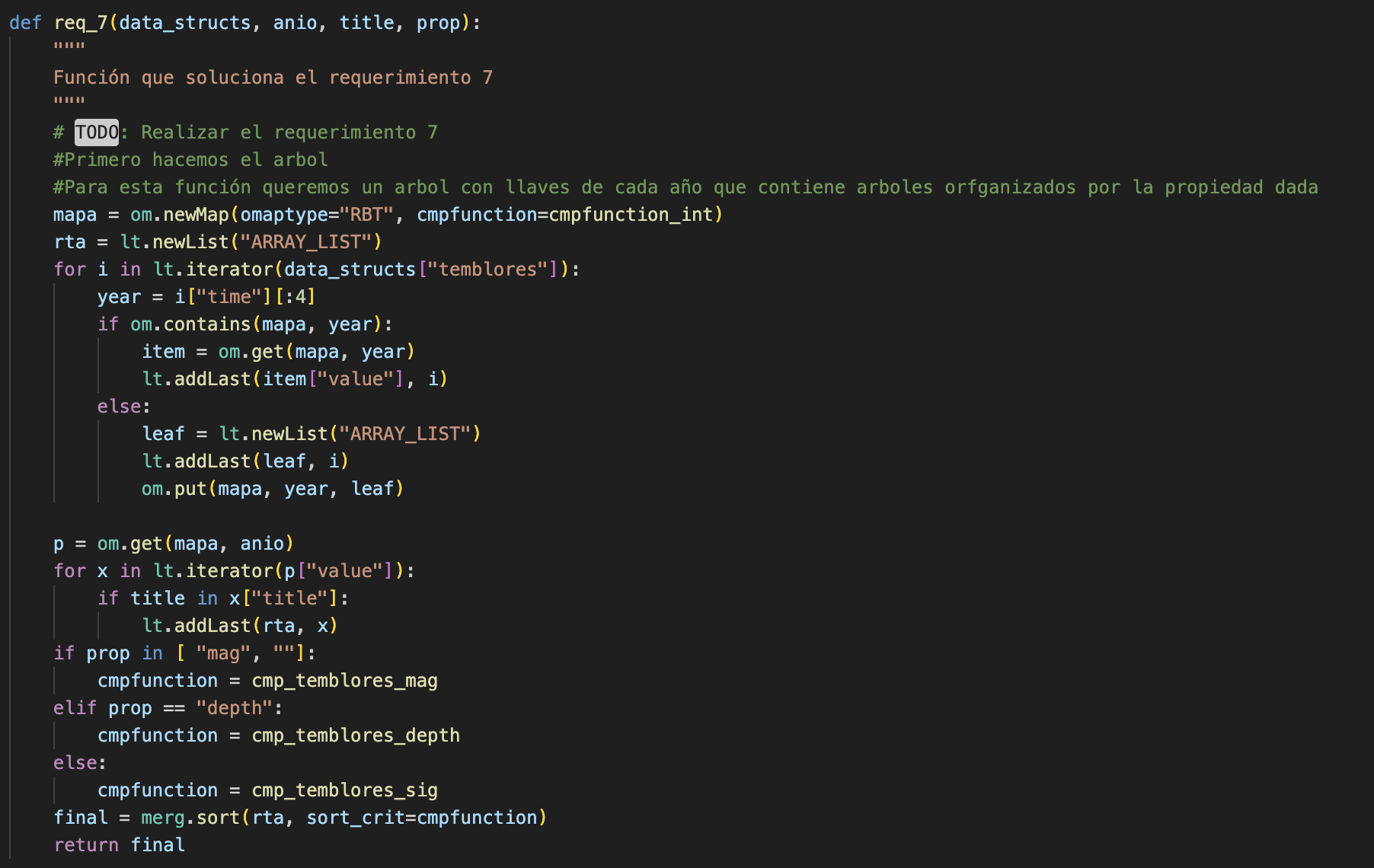
## **Análisis**

Aunque la complejidad indica que tiene complejidad O(Log(n)\*n3), la n de la iteración por la lista y la n de organizar la lista no son iguales a la n del tamaño completo del archivo. Esto hace que el incremento causado por el n3 no sea tan grande como se espere.

# **Requerimiento <<7>>**

Plantilla para el documentar y analizar cada uno de los requerimientos.

## **Descripción**



Esta función se encarga de conseguir los terremotos de una región en un año y hacer un histograma de la magnitud, la significancia o la profundidad. Primero hacemos un arbol de los años en el cual el valor de cada año es una lista con todos los terremotos de ese año. Luego, iteramos en la lista del año deseado consiguiendo los datos que corresponden a la región. Finalmente, hacemos MergeSort de acuerdo a la propiedad que quiera visualizar el usuario.

|  |  |
| --- | --- |
| **Entrada** | * Control * Año * Región * Propiedad |
| **Salidas** | * Una lista con todos los terremotos de la regíon y del año |
| **Implementado (Sí/No)** | Si se implemento, fue implementado por Andrés Tello |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 (Hacer la estructura) | O(n) |
| Paso 2 (Iterar en la lista del año) | O(m) (Donde m < n) |
| Paso 3 (MergeSort) | O(Log(n)\*n |
| ***TOTAL*** | ***O(Log(n)\*n2\*m)*** |

## **Pruebas Realizadas**

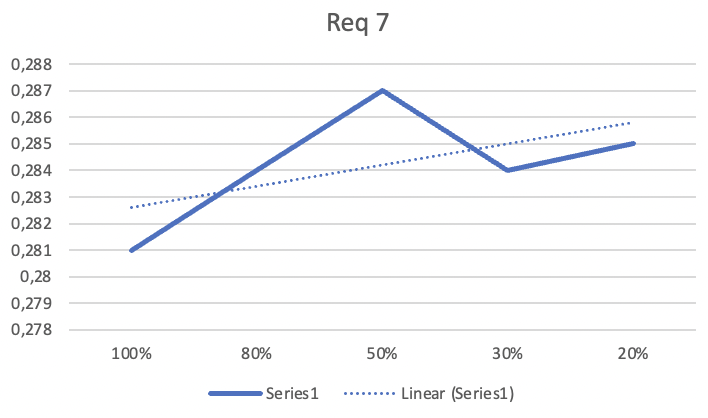
Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

|  |  |
| --- | --- |
| Procesadores | Apple M1 |
| Memoria RAM | 16 GB |
| Sistema Operativo | MacOs Monterrey |

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| large | 0.281 |
| 80% | 0.284 |
| 50% | 0.287 |
| 30% | 0.284 |
| 20% | 0.285 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

Esta función nos da una complejidad que aparenta ser alta. Sin embargo, debido a que m solo corresponde a los terremotos de un año entonces m es una fracción muy pequeña de n. Además, debido a que la estructura es un arbol RBT el arbol se mantiene muy eficiente a medida de que uno agrega elementos. Es posible que sea debido a esto que el tiempo disminuye a medida de que incrementa el porcentaje del archivo.