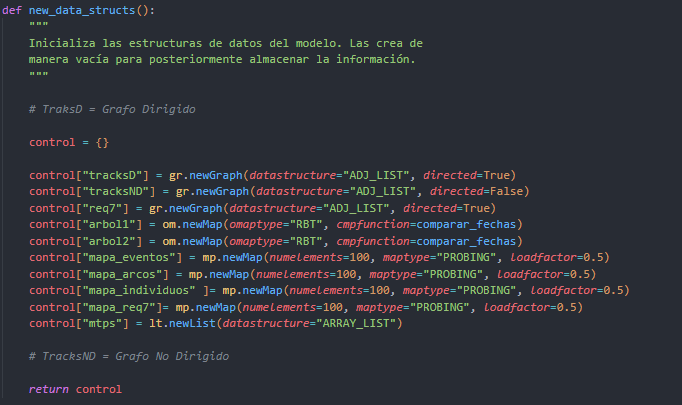
**ANÁLISIS DEL RETO**

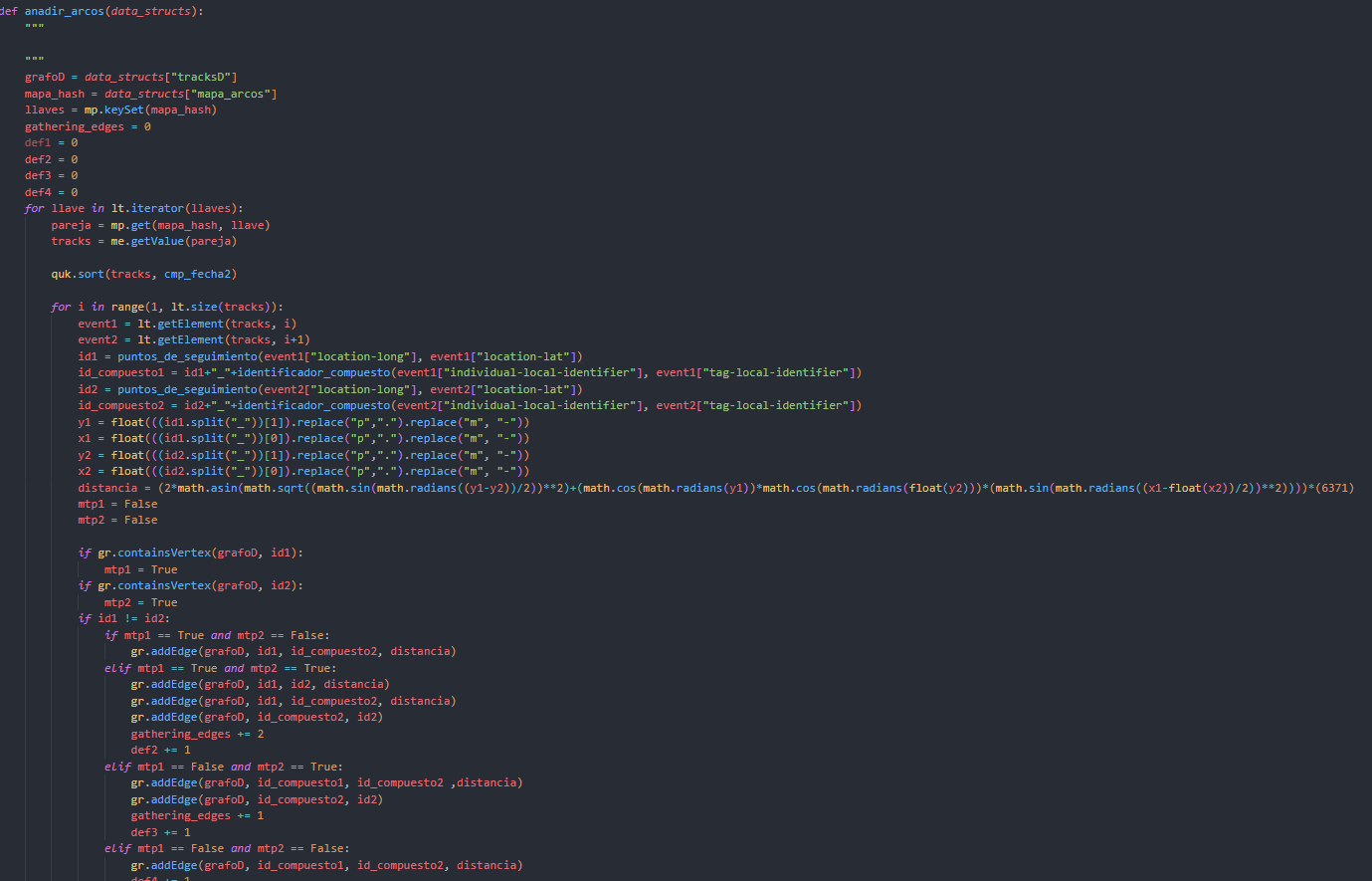
*Mateo López, 202220119, m.lopez112@uniandes.edu.co*

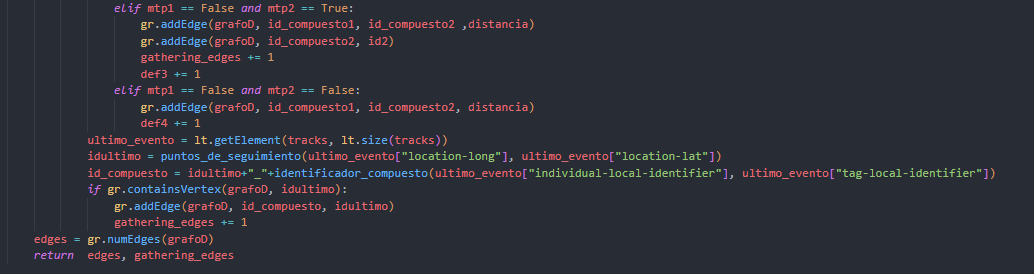
*Juan Pablo Castro, 202222086, jp.castrob1@uniandes.edu.co*

*Santiago Casasbuenas, 202214932, s.casasbuenas@uniandes.edu.co*

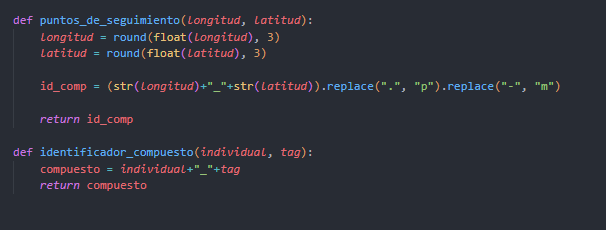
# **Carga de Datos**











## **Descripción**

Se inicia la estructura de datos con un grafo dirigido, uno no dirigido, varios mapas, dos árboles y una lista. Se añade la información de los mapas con la cual se determina el recorrido de cada lobo, también se añade cada evento a un árbol RBT. Con dicha información se añaden los vértices y los arcos a cada grafo. Luego se toman los dato de la cantidad de vertices, arcos y MTPs y se despliega.

| **Entrada** | path del archivo |
| --- | --- |
| **Salidas** | numeros de arcos y vertices, MTPs y arcos MTPs, y cinco primeros y ultimos |
| **Implementado (Sí/No)** | sí. Juan Pablo Castro |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

| **Pasos** | **Complejidad** |
| --- | --- |
| Iniciar estructura de datos | O(1) |
| Añadir información a mapas | O(N) |
| Añadir información a árbol | **O(NlogN)** |
| Añadir vértices a grafos | O(V) |
| Añadir arcos a grafos | O(E) |
| Obtener numero de vertices, MTPs, y arcos | O(V+E) |
|  | O() |
|  | O() |
|  | O() |
| ***TOTAL*** | ***O(NlogN)*** |

## **Pruebas Realizadas**

De Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos ingresados fueron “2014” y sector 6.

| Procesadores | Ryzen 3 series 5000 |
| --- | --- |
| Memoria RAM | 8GB |
| Sistema Operativo | Windows 10 |

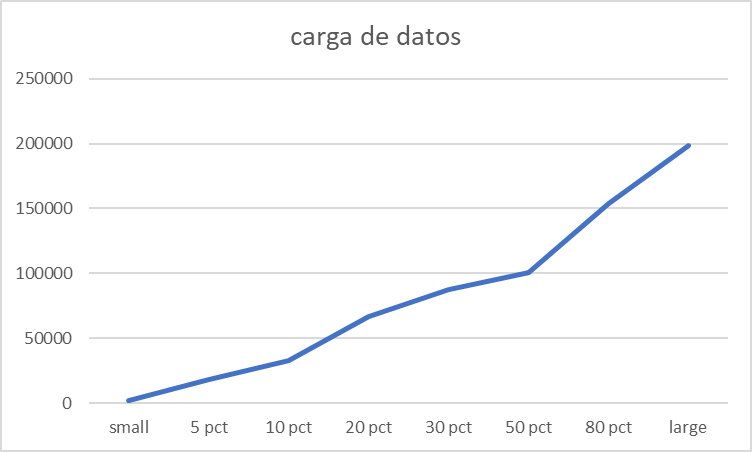
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

| **Muestra** | **Memoria** | **Tiempo (ms)** |
| --- | --- | --- |
| small |  | 1657.43 |
| 5 pct |  | 18492.30 |
| 10 pct |  | 32439.43 |
| 20 pct |  | 66437.44 |
| 30 pct |  | 87646.31 |
| 50 pct |  | 100842.43 |
| 80 pct |  | 153833.97 |
| large |  | 198520.30 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

La carga de datos tiene un comportamiento más linear de lo esperado. Este se debe a que la parte del arbol no tiene tanto efecto como el resto de los pasos, ya que son muchos y la mayoria de complejidad N.

# **Requerimiento <<1>>**

## **Descripción**

| **Entrada** |  |
| --- | --- |
| **Salidas** |  |
| **Implementado (Sí/No)** |  |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

| **Pasos** | **Complejidad** |
| --- | --- |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
| ***TOTAL*** | ***O()*** |

## **Pruebas Realizadas**

De Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos ingresados fueron “2014” y sector 6.

| Procesadores |  |
| --- | --- |
| Memoria RAM |  |
| Sistema Operativo | Windows 10 |

| **Entrada** | **Tiempo (ms)** |
| --- | --- |
| small | 4.5315999 |
| 5 pct | 17.470499 |
| 10 pct | 132.4891 |
| 20 pct | 153.5262 |
| 30 pct | 660.9092 |
| 50 pct | 1011.00070 |
| 80 pct | 1688.3006 |
| large | 2079.59560 |
| **Entrada** | **Memoria (kb)** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

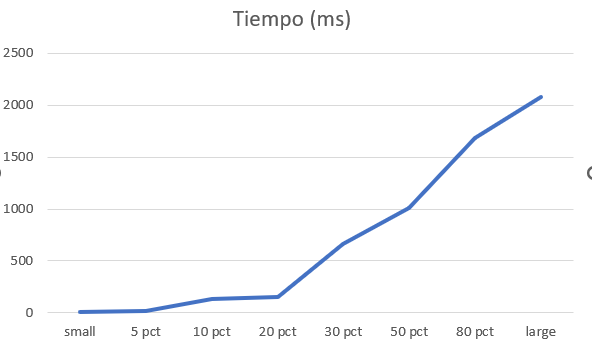
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

| **Muestra** | **Tiempo (ms)** |
| --- | --- |
| small | 4.5315999 |
| 5 pct | 17.470499 |
| 10 pct | 132.4891 |
| 20 pct | 153.5262 |
| 30 pct | 660.9092 |
| 50 pct | 1011.00070 |
| 80 pct | 1688.3006 |
| large | 2079.59560 |

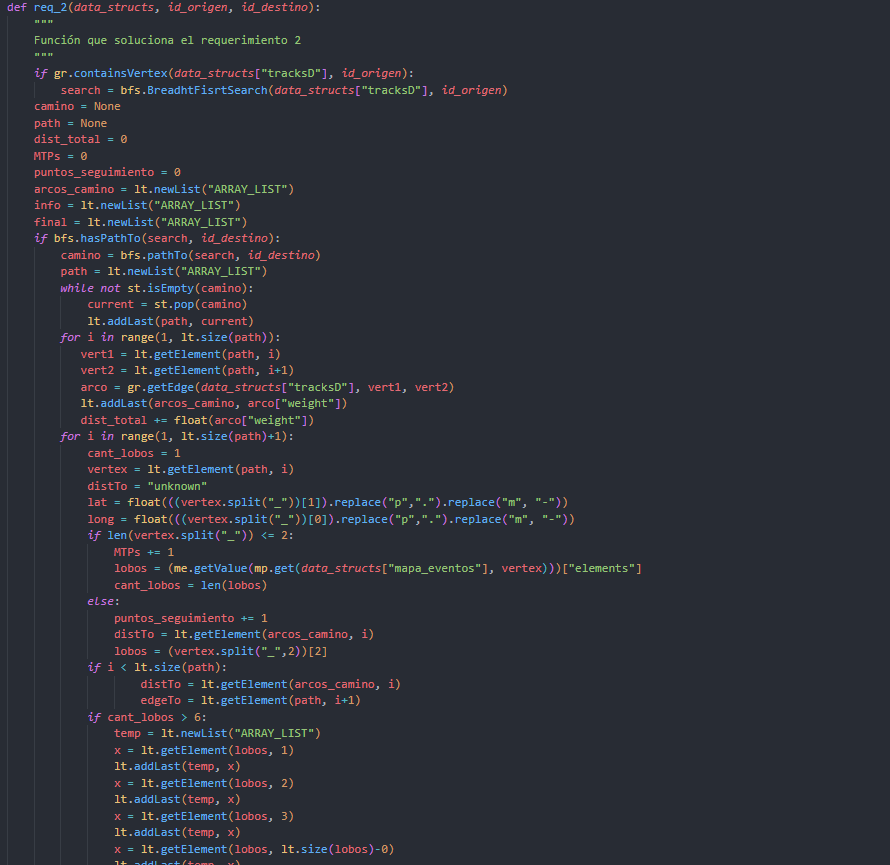
### **Graficas**

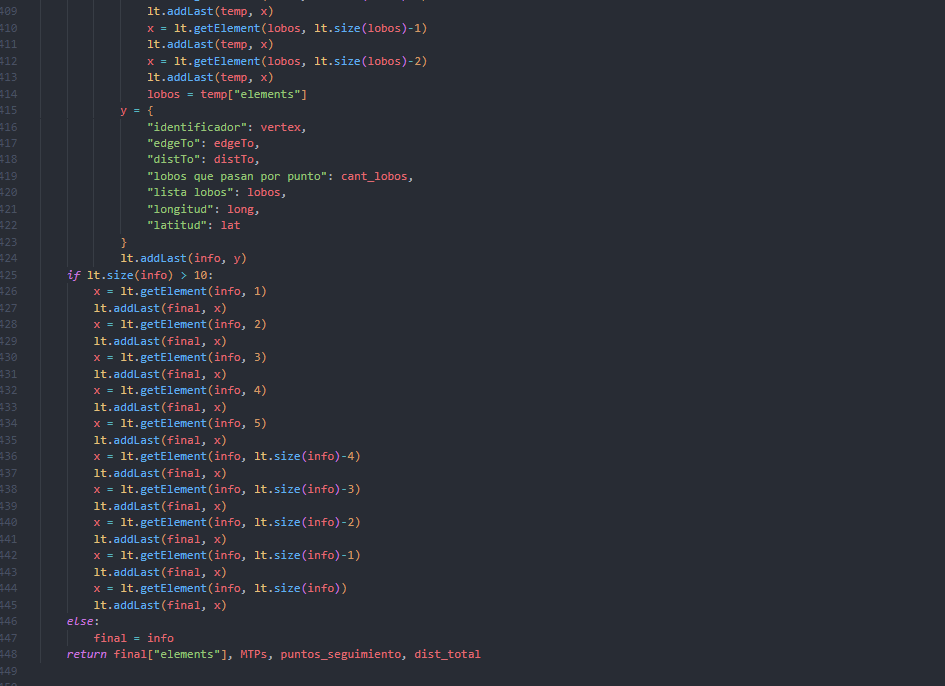
Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

# **Requerimiento <<2>>**





## **Descripción**

Se hace BFS sobre el grafo con el vertice uno como origen. Luego se saca el path a el vertice 2, y con esa pila se saca la información necesario, incluyendo la distancia total y los vertices recorridos.

| **Entrada** | grafo, id origen, id destino |
| --- | --- |
| **Salidas** | camino de id origen a id destino |
| **Implementado (Sí/No)** | Sí, Juan Pablo Castro |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

| **Pasos** | **Complejidad** |
| --- | --- |
| BFS sobre el grafo | **O(V+E)** |
| sacar el camino de id1 a id2 | O(N) |
| Limpiar datos finales | O(1) |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
| ***TOTAL*** | ***O(V+E)*** |

## **Pruebas Realizadas**

De Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos ingresados fueron “2014” y sector 6.

| Procesadores | Ryzen 3 series 5000 |
| --- | --- |
| Memoria RAM | 8GB |
| Sistema Operativo | Windows 10 |

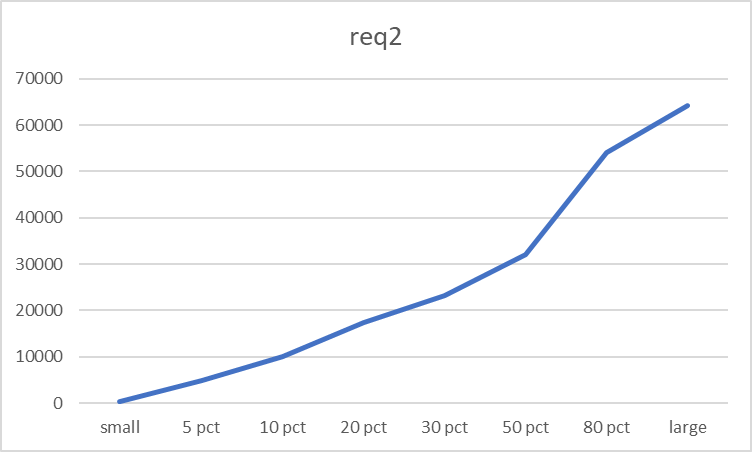
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

| **Muestra** | **Memoria** | **Tiempo (ms)** |
| --- | --- | --- |
| small |  | 374.78 |
| 5 pct |  | 4778.51 |
| 10 pct |  | 9979.57 |
| 20 pct |  | 17330.40 |
| 30 pct |  | 23110.37 |
| 50 pct |  | 31990.18 |
| 80 pct |  | 54072.16 |
| large |  | 64234.10 |

### **Graficas**

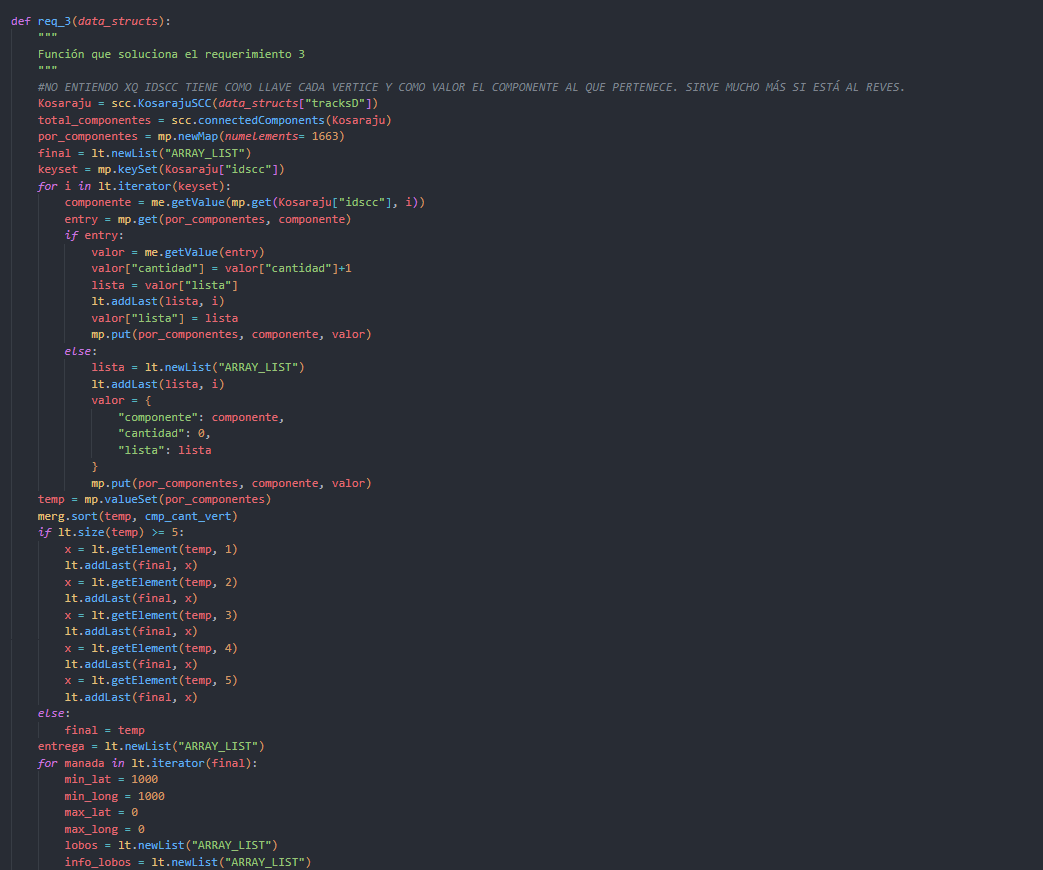
Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

El requerimiento tiene un comportamiento muy lineal, como se esperaba.

# **Requerimiento <<3>>**





Se hace Kosaraju sobre el grafo para encontrar los componentes fuertemente conectados. Luego se recorre idscc y se pone la información en un mapa ordenado por idscc. Luego se toman los componentes más grandes y se toma la información de dichos componentes.

## **Descripción**

| **Entrada** | Control |
| --- | --- |
| **Salidas** | Top 5 manadas |
| **Implementado (Sí/No)** | Sí, Juan Pablo Castro |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

| **Pasos** | **Complejidad** |
| --- | --- |
| Kosaraku | O(v+e) |
| Poner datos en mapa | O(N) |
| ordenar datos | O(NlogN) |
| tomar 5 primeros y sacar información | O(1) |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
| ***TOTAL*** | ***O(NlogN)*** |

## **Pruebas Realizadas**

De Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos ingresados fueron “2014” y sector 6.

| Procesadores | Ryzen 3 series 5000 |
| --- | --- |
| Memoria RAM | 8GB |
| Sistema Operativo | Windows 10 |

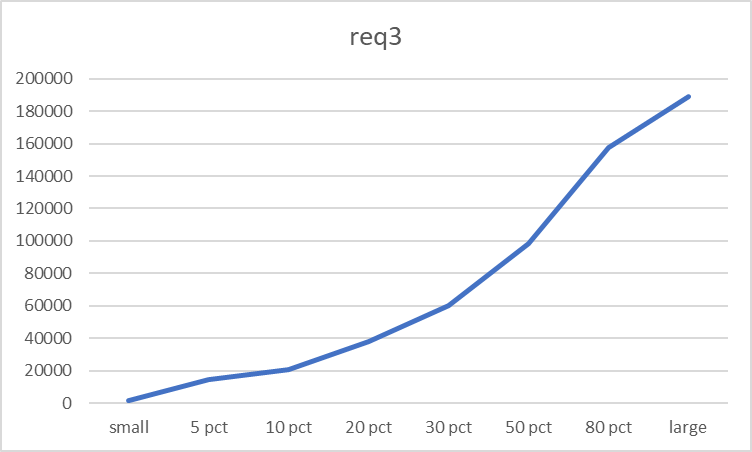
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

| **Muestra** | **Memoria** | **Tiempo (ms)** |
| --- | --- | --- |
| small |  | 1667.22 |
| 5 pct |  | 14827.33 |
| 10 pct |  | 20837.98 |
| 20 pct |  | 37995.44 |
| 30 pct |  | 59887.90 |
| 50 pct |  | 98405.88 |
| 80 pct |  | 15774.61 |
| large |  | 18873.12 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

El requerimiento tiene un comportamiento NlogN, como se esperaba.

# **Requerimiento <<4>>**

## **Descripción**

| **Entrada** | Localización geográfica del punto de origen (longitud y latitud) Localización geográfica del punto de destino (longitud y latitud) |
| --- | --- |
| **Salidas** | La distancia entre el punto GPS de origen y el punto de encuentro más cercano.  La distancia el punto de encuentro de destino más cercano y el punto GPS de destino.  La distancia total que tomará el recorrido entre los puntos de encuentro de origen y destino.  El total de puntos de encuentro que pertenecen al camino identificado (nodos).  El total de individuos/lobos distintos que utilizan el corredor identificado.  El total de segmentos que conforman la ruta identificada (arcos). Los tres primeros y tres últimos puntos de encuentro (incluyendo el punto de origen y de destino) que pertenecen al corredor identificado con la siguiente información:  El identificador del punto de encuentro.  La longitud y latitud del punto.  El número de individuos (lobos) que transitan por ese punto.  Listado con los tres primeros y tres últimos identificadores de lobos que transitan por el punto.  La distancia al siguiente punto de encuentro |
| **Implementado (Sí/No)** | Si |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

| **Pasos** | **Complejidad** |
| --- | --- |
| Implementación de función haversine | O(N) |
| Implementación Dijkstra | O(V^2) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| ***TOTAL*** | ***O(V^2)*** |

## **Pruebas Realizadas**

De Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos ingresados fueron “2014” y sector 6.

| Procesadores | Intel core i7 8th gen |
| --- | --- |
| Memoria RAM | 32 GB |
| Sistema Operativo | Windows 10 |

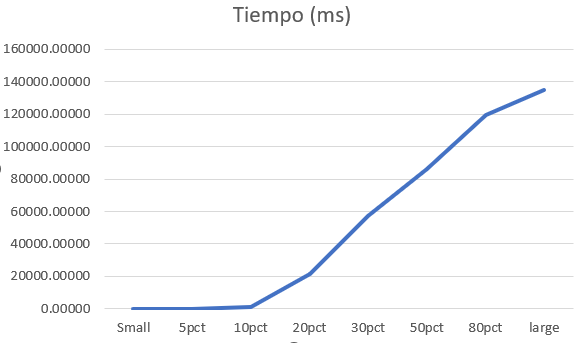
| **Entrada** | **Tiempo (ms)** |
| --- | --- |
| Small | 34.16040 |
| 5pct | 202.88740 |
| 10pct | 1405.20589 |
| 20pct | 21300.17770 |
| 30pct | 57380.15359 |
| 50pct | 86001.42950 |
| 80pct | 119327.76400 |
| large | 134635.71930 |
| **Entrada** | **Memoria (kb)** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

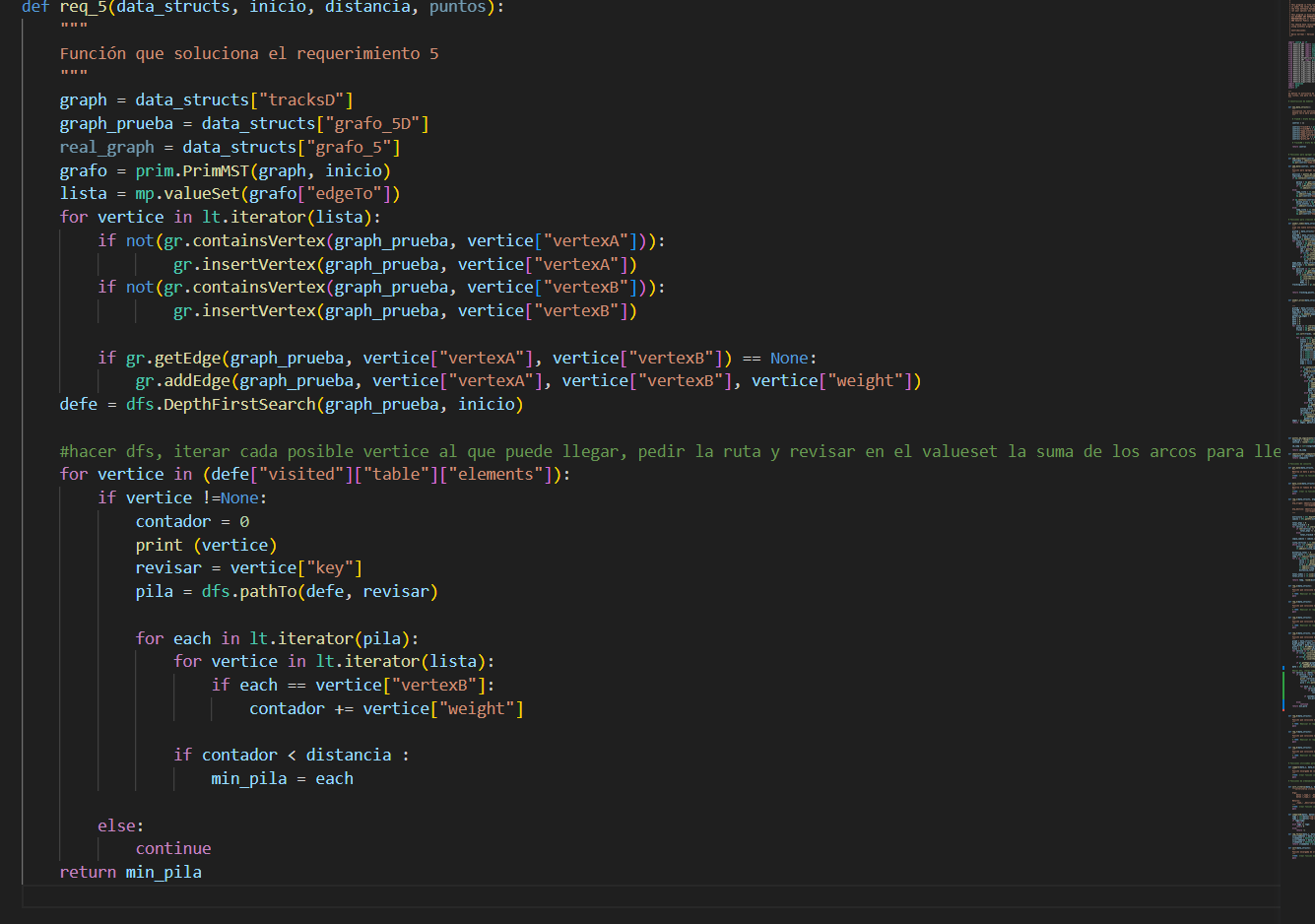
| **Muestra** | **Tiempo (ms)** |
| --- | --- |
| small | 34.16040 |
| 5 pct | 202.88740 |
| 10 pct | 1405.20589 |
| 20 pct | 21300.17770 |
| 30 pct | 57380.15359 |
| 50 pct | 86001.42950 |
| 80 pct | 119327.76400 |
| large | 134635.71930 |

### **Graficas**



## **Análisis**

# **Requerimiento <<5>>**



## **Descripción**

Teniendo un punto de encuentro inicial, realiza un MST sobre el grafo, después se crea un grafo nuevo con la información que entrega la función de Prim y se realiza una busqueda DFS en este para encontrar el vértice con mayor distancia pero que sea menor a una dada como parámetro inicial.

| **Entrada** | Punto de encuentro inicial, distancia máxima, puntos de encuentro mínimos |
| --- | --- |
| **Salidas** | Información del vértice que cumpla con tales características |
| **Implementado (Sí/No)** | Si, Mateo López |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

| **Pasos** | **Complejidad** |
| --- | --- |
| Realizar un MST usando el algoritmo de PRIM | O ( ( V + E ) l o g V ) |
| Añadir vértices al grafo | O(V) |
| Añadir arcos al grafo | O(E) |
| Realizar un DFS en el grafo nuevo | O(V) |
| Para cada vértice encontrado en el recorrido DFS, encontrar el camino mediante path To | O(V) |
| Para cada vértice en el recorrido encontrado en el paso anterior, encontrar la suma de los arcos desde el mtp original hasta tal vértice | O(V^3) |
| Revisa que la suma de pesos sea menor que el máximo y mayor a la anterior mayor suma. Si lo es, cambiar el valor a entregar por el nombre del vértice con tal peso | O(1) |
| ***TOTAL*** | ***O(V^3)*** |

## **Pruebas Realizadas**

De Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos ingresados fueron “2014” y sector 6.

| Procesadores |  |
| --- | --- |
| Memoria RAM |  |
| Sistema Operativo | Windows 10 |

| **Entrada** | **Tiempo (ms)** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Entrada** | **Memoria (kb)** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

| **Muestra** | **Memoria** | **Tiempo (ms)** |
| --- | --- | --- |
| small |  |  |
| 5 pct |  |  |
| 10 pct |  |  |
| 20 pct |  |  |
| 30 pct |  |  |
| 50 pct |  |  |
| 80 pct |  |  |
| large |  |  |

### **Graficas**

Debido a que no se pudo terminar con éxito el requerimiento, las tomas de datos no pudieron ser realizadas. Sin embargo, por tener una gran cantidad de ciclos concatenados, se esperaría que el tiempo para completar el requerimiento tenga un crecimiento cúbico.

## **Análisis**

# **Requerimiento <<6>>**

## **Descripción**

| **Entrada** |  |
| --- | --- |
| **Salidas** |  |
| **Implementado (Sí/No)** |  |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

| **Pasos** | **Complejidad** |
| --- | --- |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
| ***TOTAL*** | ***O()*** |

## **Pruebas Realizadas**

De Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos ingresados fueron “2014” y sector 6.

| Procesadores |  |
| --- | --- |
| Memoria RAM |  |
| Sistema Operativo | Windows 10 |

| **Entrada** | **Tiempo (ms)** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Entrada** | **Memoria (kb)** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

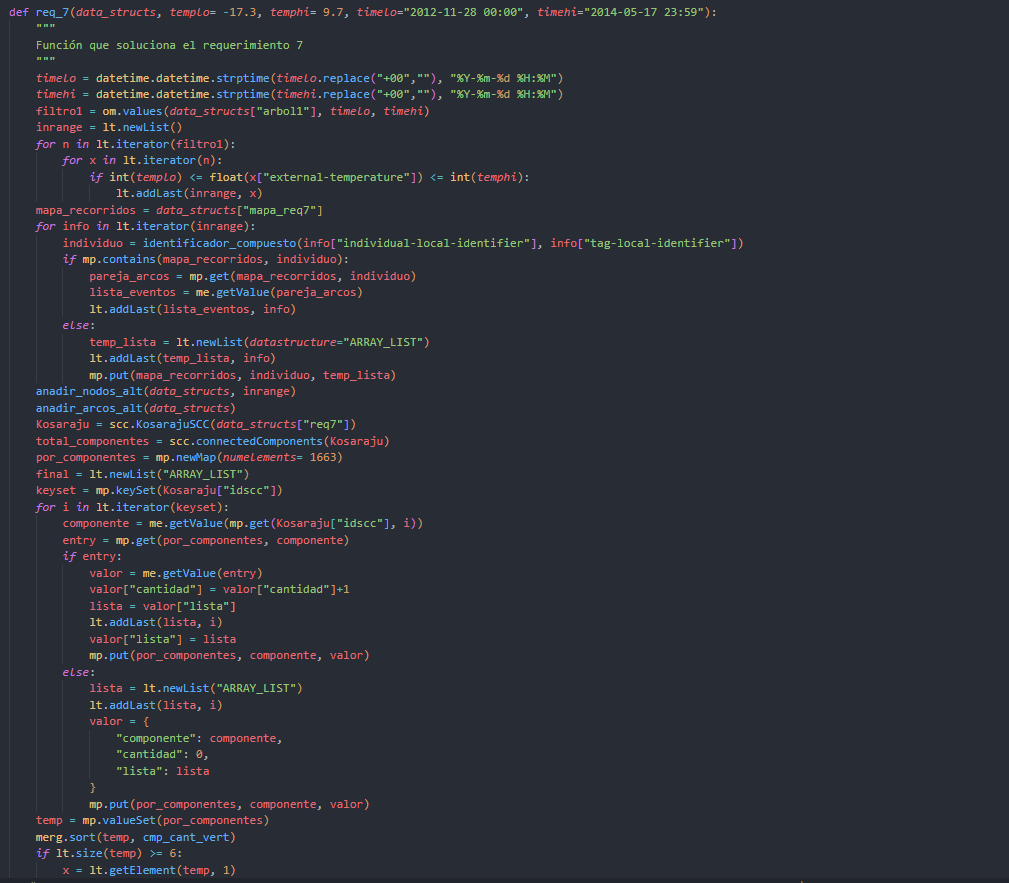
| **Muestra** | **Memoria** | **Tiempo (ms)** |
| --- | --- | --- |
| small |  |  |
| 5 pct |  |  |
| 10 pct |  |  |
| 20 pct |  |  |
| 30 pct |  |  |
| 50 pct |  |  |
| 80 pct |  |  |
| large |  |  |

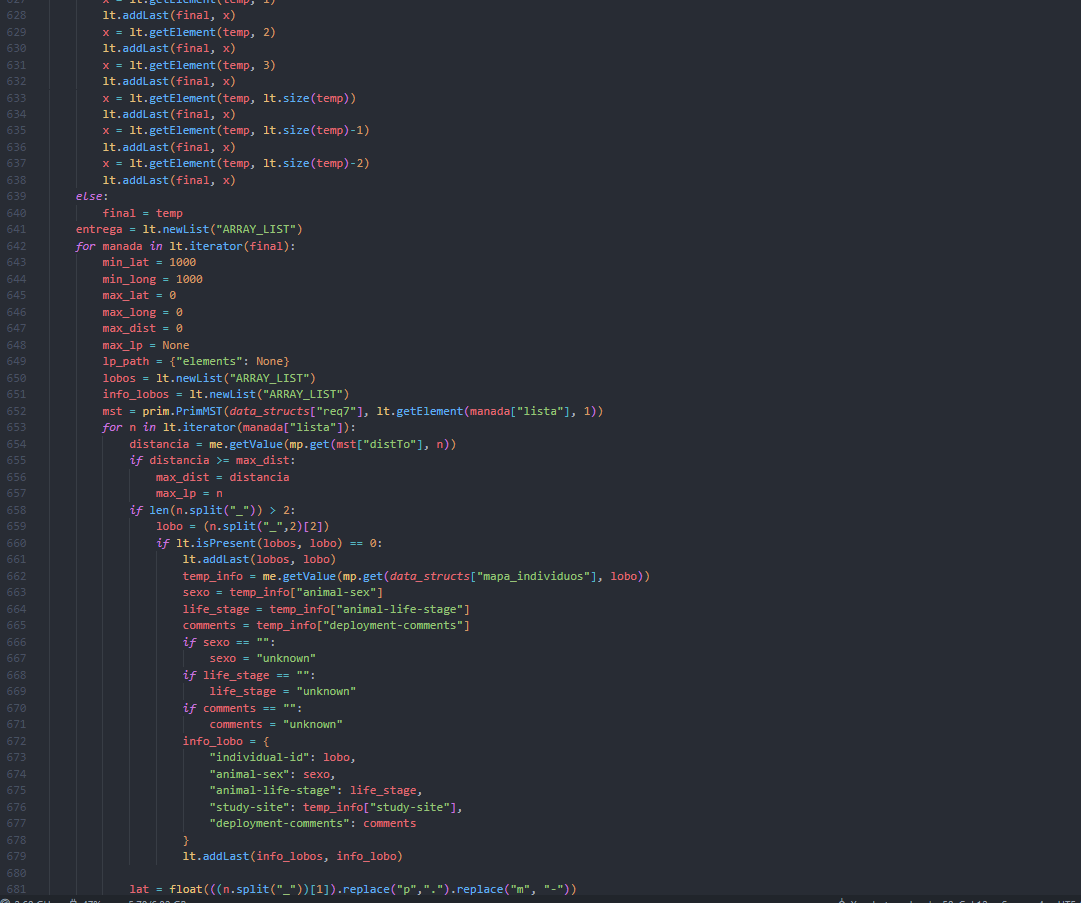
### **Graficas**

Las gráficas con la representación de las pruebas realizadas.

## **Análisis**

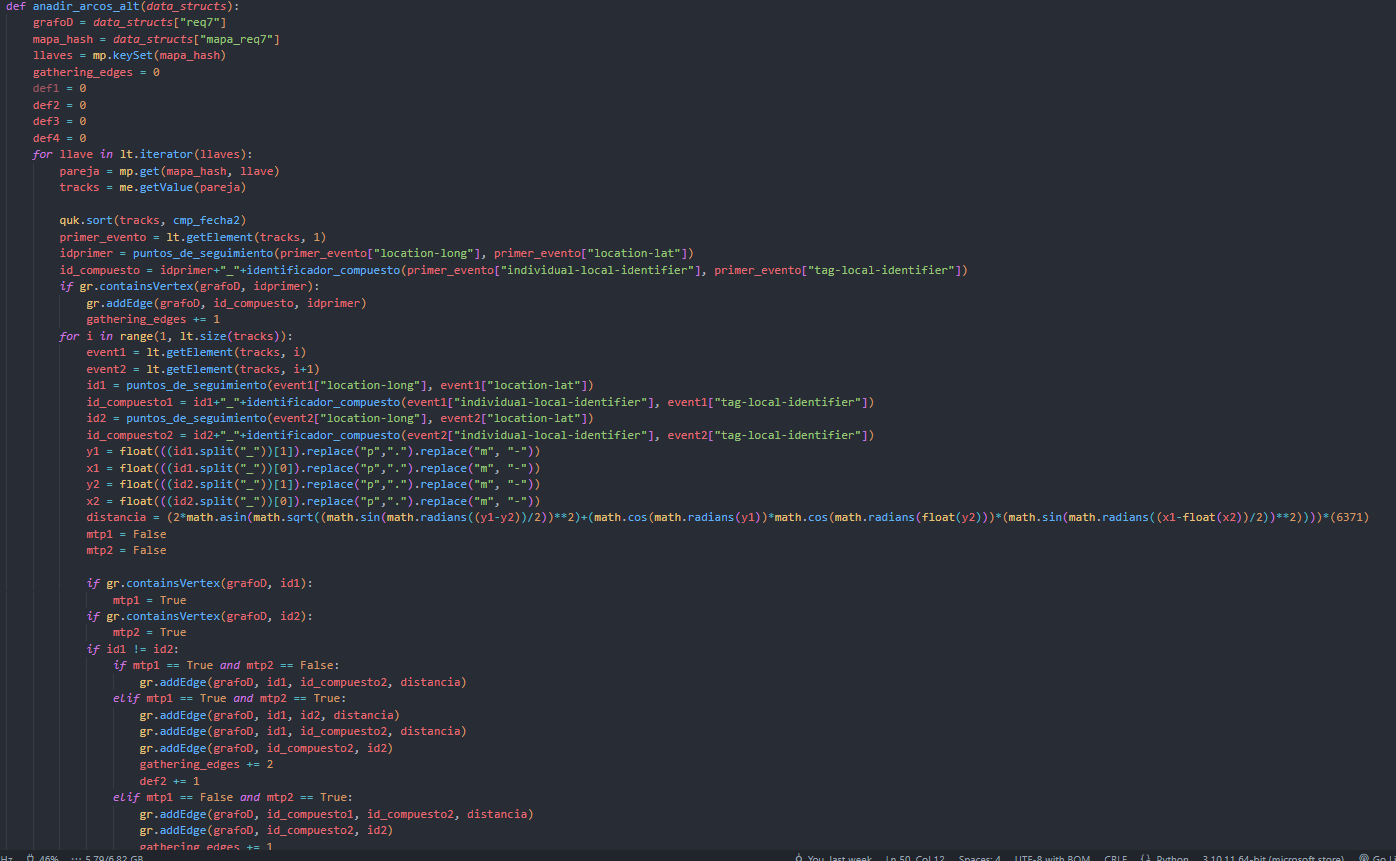
# **Requerimiento <<7>>**













## **Descripción**

En un arbol que estaba ordenador por timestamp, se saca el rango requerido, sí la información también está en el rango de temperaturas, se añade a un nuevo grafo, el cual se construye desde cero, creando los mismos mapas de la carga de datos, pero solo con la información del rango. Luego se hace Kosaraju sobre dicho grafo y se hace toma la info de los 3 primeros y 3 ultimos con más vertices. Para esos componentes se busca el camino más largo dentro del componente con MST.

| **Entrada** | grafo, rango de fechas, rango de temperatura |
| --- | --- |
| **Salidas** | SCC con info y LP con info |
| **Implementado (Sí/No)** | Sí, Juan Pablo Castro |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

| **Pasos** | **Complejidad** |
| --- | --- |
| Filtrar el arbol | O(N) |
| crear grafo | O(V+E) |
| Kosaraju | O(V+E) |
| MST | O(V+E) |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
| ***TOTAL*** | ***O(V+E)*** |

## **Pruebas Realizadas**

De Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos ingresados fueron “2014” y sector 6.

| Procesadores | Ryzen 3 series 5000 |
| --- | --- |
| Memoria RAM | 8GB |
| Sistema Operativo | Windows 10 |

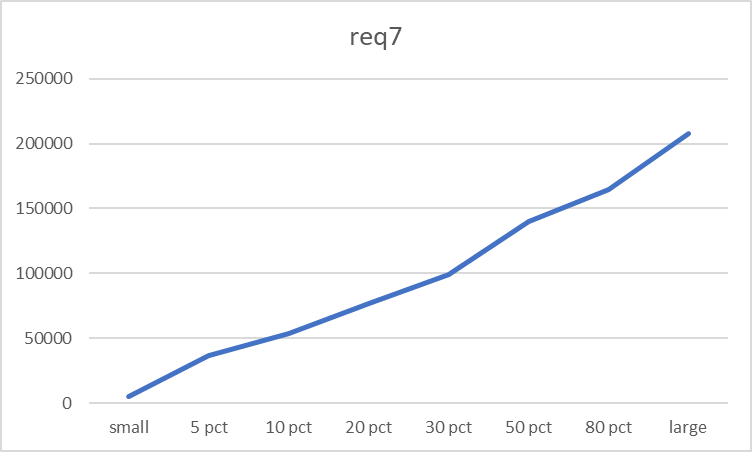
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

| **Muestra** | **Memoria** | **Tiempo (ms)** |
| --- | --- | --- |
| small |  | 4985.32 |
| 5 pct |  | 36944.32 |
| 10 pct |  | 53677.28 |
| 20 pct |  | 76994.90 |
| 30 pct |  | 98882.61 |
| 50 pct |  | 13988.69 |
| 80 pct |  | 16477.81 |
| large |  | 207624.19 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

Comportamiento linear, sin embargo, como el punto de corte es tan alto, se sigue demorando bastante.

# **Requerimiento <<8>>**

## **Descripción**

| **Entrada** |  |
| --- | --- |
| **Salidas** |  |
| **Implementado (Sí/No)** |  |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

| **Pasos** | **Complejidad** |
| --- | --- |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
|  | O() |
| ***TOTAL*** | ***O()*** |

## **Pruebas Realizadas**

De Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos ingresados fueron “2014” y sector 6.

| Procesadores |  |
| --- | --- |
| Memoria RAM |  |
| Sistema Operativo | Windows 10 |

| **Entrada** | **Tiempo (ms)** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Entrada** | **Memoria (kb)** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

| **Muestra** | **Memoria** | **Tiempo (ms)** |
| --- | --- | --- |
| small |  |  |
| 5 pct |  |  |
| 10 pct |  |  |
| 20 pct |  |  |
| 30 pct |  |  |
| 50 pct |  |  |
| 80 pct |  |  |
| large |  |  |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.

## **Análisis**