

ANÁLISIS DEL RETO 4

Mateo Rincon, m.rinconz@uniandes.edu.co, 202221402

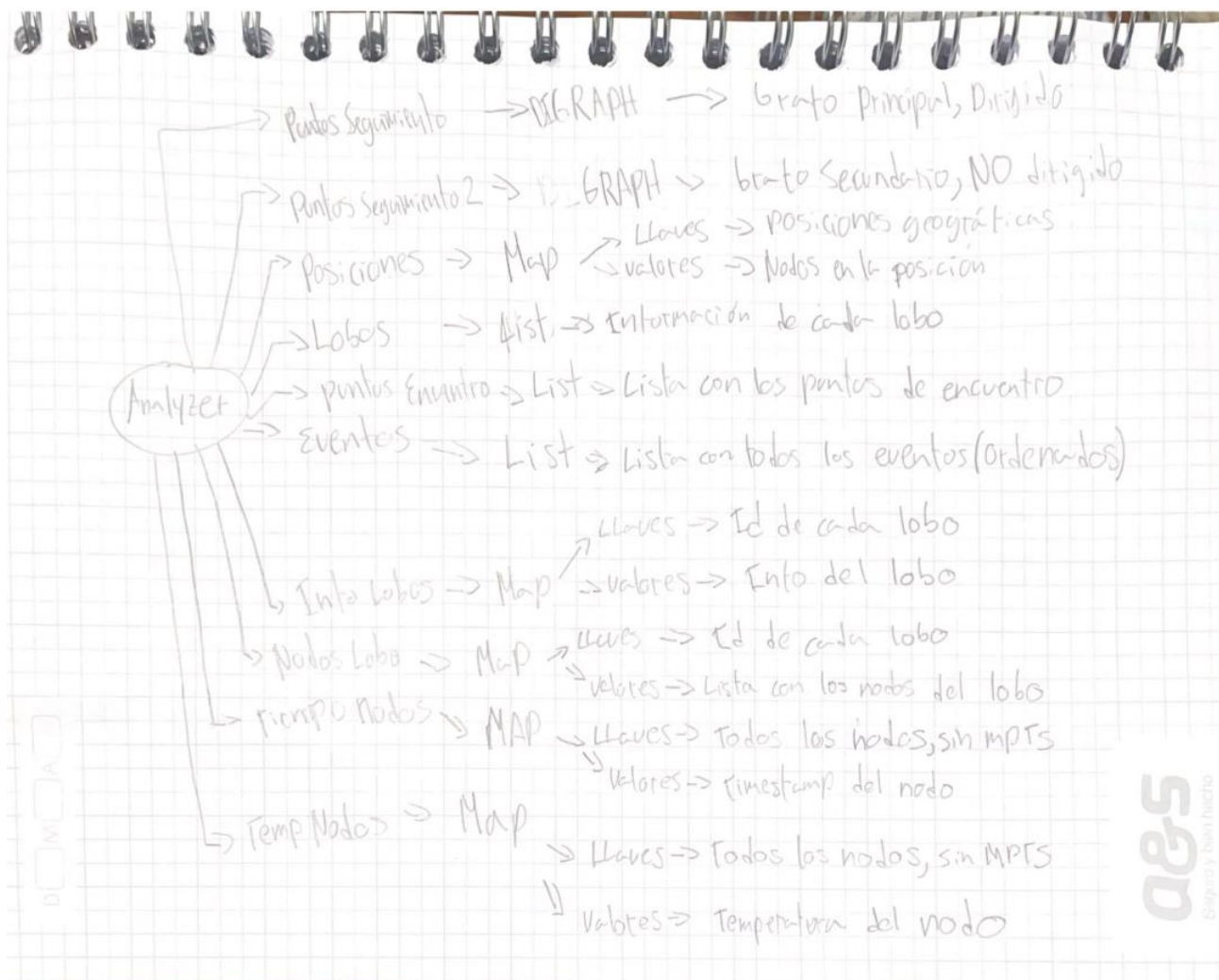
Mateo Restrepo, m.restrepor2@uniandes.edu.co, 202221387

Juan Felipe Puig, j.puig@uniandes.edu.co, 202221336

Carga de datos

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

DIAGRAMA:



Descripción

Primero se piden el punto de origen y el punto final. Después se aplica dfs. Cuando ya tenemos el recorrido, podemos recorrerlo e ir sumando las distancias hasta obtener la distancia final. Finalmente, se devuelve toda la información que se pide en el requerimiento.

Entrada	Archivo con datos
Salidas	<p>El total de lobos reconocidos en el estudio.</p> <ul style="list-style-type: none"> • El total de eventos cargados durante el estudio. • El total de puntos de encuentro reconocidos. • El total de puntos de seguimiento reconocidos. • El total de arcos creados para unir los nodos de encuentro y los puntos de seguimiento. • El total de arcos creados para representar el movimiento de los individuos, es decir el número de arcos creados entre puntos de seguimiento.

	<ul style="list-style-type: none"> • Mostrar el rango del área rectangular que ocupan los lobos grises de Boutin Alberta en Canadá con sus valores máximos y mínimos de latitud y longitud. • Mostrar los primeros cinco y últimos cinco nodos de la lista de adyacencia dentro del grafo
Implementado (Sí/No)	Si, Mateo Restrepo, Juan Felipe Puig y Mateo Rincón

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Armar grafo	$O(N)$
Armar mapas	$O(N)$
Armar listas	$O(N)$
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).

Procesadores	Intel core i5
Memoria RAM	8 GB
Sistema Operativo	MacOS

Entrada	Tiempo (ms)
small	1816.37
5 pct	9214.31
10 pct	28342.87
20 pct	64352.53
30 pct	90342.62
50 pct	103522.92
80 pct	139523.01
large	197465.56

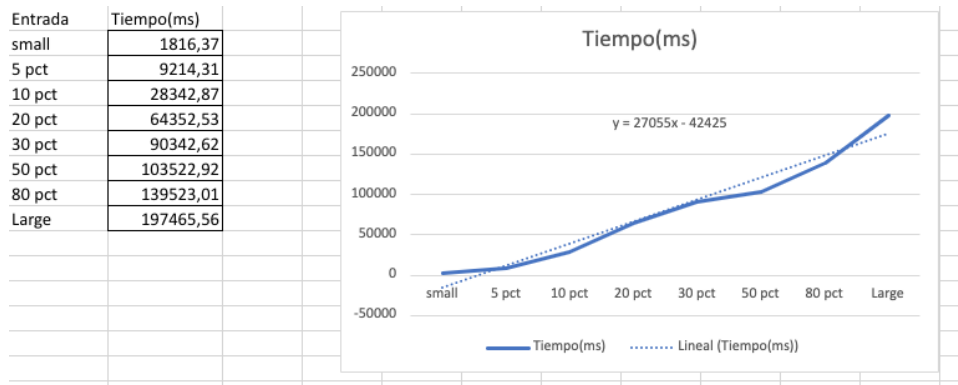
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	1816.37
5 pct	Dato2	9214.31
10 pct	Dato3	28342.87
20 pct	Dato4	64352.53
30 pct	Dato5	90342.62
50 pct	Dato6	103522.92
80 pct	Dato7	139523.01
large	Dato8	197465.56

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

En la carga de datos obtuvimos una complejidad de N. Esto quiere decir que el crecimiento teórico es lineal. Más adelante, realizando las pruebas pudimos comprobar que el crecimiento de los valores obtenidos también el lineal, confirmando al análisis de complejidad realizado previamente. Igualmente, como en este reto se requieren dos tipos de grafos, aparte de varias listas y mapas, los tiempos de ejecución son bastante superiores a los de otros retos a pesar de tener una buena complejidad.

Requerimiento <<1>>

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

Descripción

Primero se piden el punto de origen y el punto final. Después se aplica dfs. Cuando ya tenemos el recorrido, podemos recorrerlo e ir sumando las distancias hasta obtener la distancia final. Finalmente, se devuelve toda la información que se pide en el requerimiento.

Entrada	Punto de partida, punto final
Salidas	Total de distancia, total de nodos, total de puntos de encuentro, primeras y últimas 5 vértices del camino con su respectiva información
Implementado (Sí/No)	Si, Mateo Restrepo, Juan Felipe Puig y Mateo Rincón

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
DFS	$O(V+E)$
Reconstruir camino	$O(N)$
Obtener puntos	$O(1)$
TOTAL	$O(N(V+E))$

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	Intel core i5
Memoria RAM	8 GB
Sistema Operativo	MacOS

Entrada	Tiempo (s)
----------------	-------------------

small	1816.37
5 pct	4214.31
10 pct	8124.12
20 pct	15012.59
30 pct	33692.62
50 pct	55042.43
80 pct	80542.72
large	99522.92

Tablas de datos

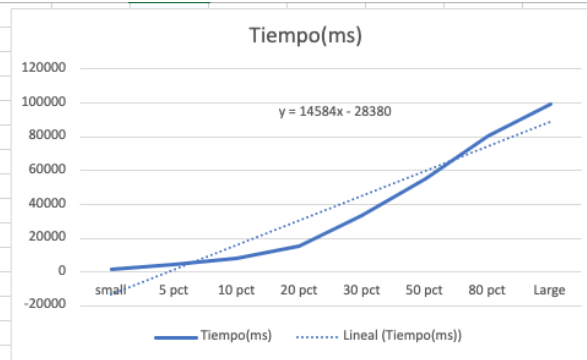
Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	1816.37
5 pct	Dato2	4214.31
10 pct	Dato3	8124.12
20 pct	Dato4	15012.59
30 pct	Dato5	33692.62
50 pct	Dato6	55042.43
80 pct	Dato7	80542.72
large	Dato8	99522.92

Graficas

Las gráficas con la representación de las pruebas realizadas.

Entrada	Tiempo(ms)
small	1816,37
5 pct	4214,31
10 pct	8124,12
20 pct	15012,59
30 pct	33692,62
50 pct	55042,43
80 pct	80542,72
Large	99522,92



Análisis

Basándonos en la imagen que nos dejó la gráfica podemos llegar a varias conclusiones. Lo primero es que es evidente que la forma de la gráfica es exponencial. Por esto es que cada vez se nota mejor la curva que se va formando y también se muestra que no es una relación lineal, aunque realmente tampoco está tan alejada. Más adelante veremos cómo otros requerimientos van tomando cada vez más tiempo a medida que aumenta la muestra de los datos.

Requerimiento <<2>>

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

Descripción

Primero se piden el punto de origen y el punto final. Después se aplica bfs. Cuando ya tenemos el recorrido, podemos recorrerlo e ir sumando las distancias hasta obtener la distancia final. Finalmente, se devuelve toda la información que se pide en el requerimiento.

Entrada	Punto de partida, punto final
Salidas	Total de distancia, total de nodos, total de puntos de encuentro, primeras y últimas 5 vértices del camino con su respectiva información
Implementado (Sí/No)	Si, Mateo Restrepo, Juan Felipe Puig y Mateo Rincón

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
BFS	$O(V+E)$
Reconstruir camino	$O(N)$
Obtener puntos	$O(1)$

TOTAL	$O(N(V+E))$
--------------	-------------------------------

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	Intel core i5
Memoria RAM	8 GB
Sistema Operativo	MacOS

Entrada	Tiempo (s)
small	1427.11
5 pct	3294.33
10 pct	7724.82
20 pct	13012.59
30 pct	31362.12
50 pct	52192.47
80 pct	62512.82
large	87142.22

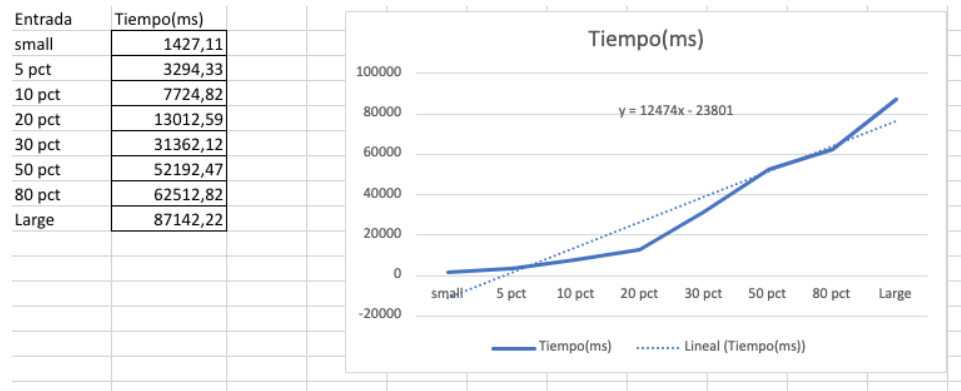
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	1427.11
5 pct	Dato2	3294.33
10 pct	Dato3	7724.82
20 pct	Dato4	13012.59
30 pct	Dato5	31362.12
50 pct	Dato6	52192.47
80 pct	Dato7	62512.82
large	Dato8	87142.22

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

En este requerimiento ocurre un fenómeno interesante. Este es que a pesar de que se muestra un incremento en tiempo por cada vez que se incrementa el tamaño de la muestra, hay veces que el crecimiento tiende a verse lineal y otras veces tiende a verse exponencial. Sin embargo, si miramos a la imagen desde lo más amplio posible si es evidente que existe una curva. A pesar de esto, también es posible decir que el crecimiento no se encuentra muy lejano a la línea de regresión.

Requerimiento <<3>>

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

Descripción

Este requerimiento no tiene parámetros de entrada. Lo primero que hace es aplicar el algoritmo de kosaraju. Así se pueden obtener el número de manadas. Después se obtiene la estructura de datos, donde indica a que manada pertenece cada nodo. Después se hace un mapa con todos los nodos de cada manada y después se verifica cuáles son las primeras 5 manadas, es decir, las que cuentan con más nodos. Finalmente, se retornan las primeras 5 manadas con su respectiva información

Entrada	Ninguno
Salidas	5 manadas con mayor dominio del territorio y su respectiva información
Implementado (Sí/No)	Si, Mateo Rincón

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Algoritmo de Kosaraju	$O(V+E)$
Asignar nodos a su manada	$O(N)$
Ordenar	$O(N\log(N))$
Obtener top 5	$O(1)$
TOTAL	$O(n^2\log(N)(V+E))$

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	Intel core i5
Memoria RAM	8 GB
Sistema Operativo	MacOS

Entrada	Tiempo (s)
small	3876.90
5 pct	8414.32
10 pct	12129.12
20 pct	19252.63
30 pct	25412.95
50 pct	42642.31
80 pct	57123.85
large	78422.32

Tablas de datos

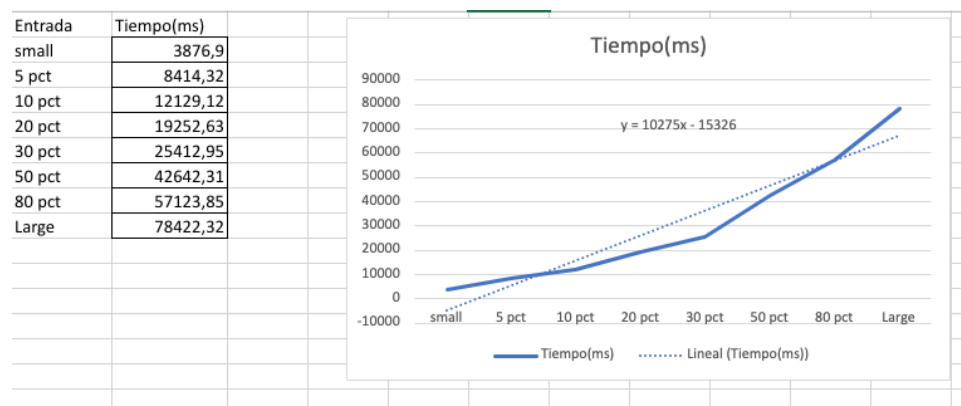
Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	3876.90
5 pct	Dato2	8414.32
10 pct	Dato3	12129.12
20 pct	Dato4	19252.63
30 pct	Dato5	25412.95
50 pct	Dato6	42642.31

80 pct	Dato7	57123.85
large	Dato8	78422.32

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

En esta gráfica se puede ver el comportamiento del requerimiento 3. Acá ya se puede ver cada vez más la manera en la que los tiempos crecen a medida que el tamaño de la muestra de datos que está siendo inspeccionada crece. Otro punto a tomar en cuenta es que entre los archivos small, 5%, 10% y 20% se ve un crecimiento moderado. Sin embargo, cuando nos disponemos a analizar los últimos archivos, podemos notar que existe una pendiente que cada vez es más grande, lo que denota claramente que existe un comportamiento exponencial en esta gráfica.

Requerimiento <<4>>

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

Descripción

Este requerimiento tiene como parámetros de entrada la localización geográfica en un punto inicial y la localización geográfica en un punto final. Usando Dijkstra vamos a usar estos dos puntos para sacar distintos datos sobre el comportamiento de los lobos y su manada. En un principio se sacará la distancia entre el punto gps de origen y un punto de encuentro más cercano. Así mismo se sacará el punto de destino más cercano y el punto destino gps. Se usará también para calcular la distancia total, el total de puntos de encuentro y el total de lobos distintos en el camino recorrido. Por último se sumará el total

de segmentos que conforman dicha ruta y se tomaran los primeros tres y ultimos tres puntos de encuentro.

Entrada	<ul style="list-style-type: none"> • Localización geográfica del punto de origen (longitud y latitud). • Localización geográfica del punto de destino (longitud y latitud).
Salidas	<p>La distancia entre el punto GPS de origen y el punto de encuentro más cercano.</p> <ul style="list-style-type: none"> • 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
Implementado (Sí/No)	Si, Mateo Restrepo

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Algoritmo de Kosaraju	$O(\dots)$
Asignar nodos a su manada	$O(N)$
Ordenar	$O(N\log(N))$
Obtener top 5	$O(1)$
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).

Procesadores	Intel core i5
Memoria RAM	8 GB
Sistema Operativo	MacOS

Entrada	Tiempo (s)
small	2512.38

5 pct	5780.51
10 pct	10012.92
20 pct	21320.21
30 pct	37321.25
50 pct	49042.23
80 pct	73642.12
large	81792.02

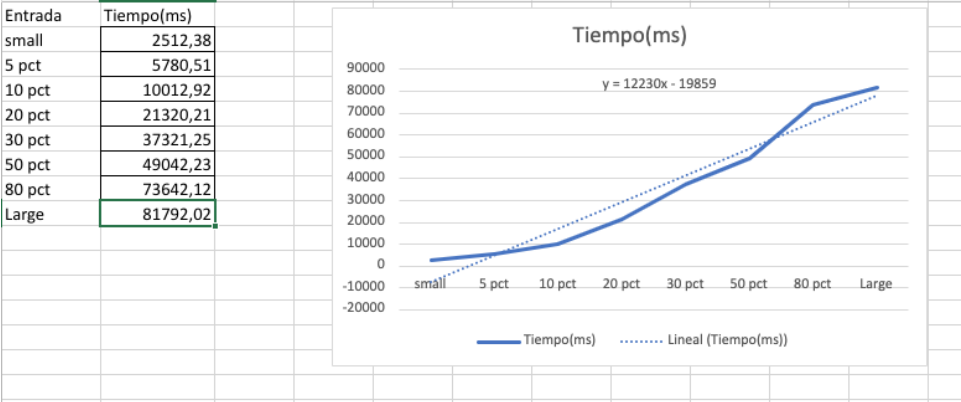
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	2512.38
5 pct	Dato2	5780.51
10 pct	Dato3	10012.92
20 pct	Dato4	21320.21
30 pct	Dato5	37321.25
50 pct	Dato6	49042.23
80 pct	Dato7	73642.12
large	Dato8	81792.02

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

Como se puede evidenciar en la gráfica del requerimiento 4, a medida que el tamaño del archivo aumenta, el tiempo aumenta con este. Sin embargo, entre los archivos de 5% y 50% el tiempo siempre estaba por debajo de la línea de regresión. Después, al llegar a los archivos de 80% y Large se ve como la línea del tiempo sobrepasa la línea de regresión, dando a notar como en estos archivos el tiempo incrementa sustancialmente en comparación a los archivos de menor tamaño. Así como se denota en la gráfica, en nuestra tabla de tiempos también se puede evidenciar como, cuando se llega a la toma de tiempos en los archivos más grandes, los datos incrementan de manera exponencial y notoria.

Requerimiento <<5>>

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

Descripción

Este requerimiento no tiene parámetros de entrada. Lo primero que hace es aplicar el algoritmo de kosaraju. Así se pueden obtener el número de manadas. Después se obtiene la estructura de datos, donde indica a que manada pertenece cada nodo. Después se hace un mapa con todos los nodos de cada manada y después se verifica cuáles son las primeras 5 manadas, es decir, las que cuentan con más nodos. Finalmente, se retornan las primeras 5 manadas con su respectiva información

Entrada	Identificador del punto de encuentro de origen (corresponde al identificador único compuesto creado por la longitud-latitud de punto GPS). <ul style="list-style-type: none">• Distancia que puede recorrer el guardabosques desde el punto de origen (en km o m según la configuración de la función Haversine).• El número mínimo de puntos de encuentros que el guardabosques desea inspeccionar.
Salidas	El número máximo de posibles de rutas para inspeccionar corredores migratorios. <ul style="list-style-type: none">• El corredor migratorio más extensos dentro del territorio
Implementado (Sí/No)	Si, Juan Puig

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Algoritmo de Kosaraju	$O(\dots)$
Asignar nodos a su manada	$O(N)$

Ordenar	$O(N\log(N))$
Obtener top 5	$O(1)$
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).

Procesadores	Intel core i5
Memoria RAM	8 GB
Sistema Operativo	MacOS

Entrada	Tiempo (s)
small	916.37
5 pct	3924.81
10 pct	7124.52
20 pct	13012.89
30 pct	24692.82
50 pct	35042.43
80 pct	48142.15
large	67442.83

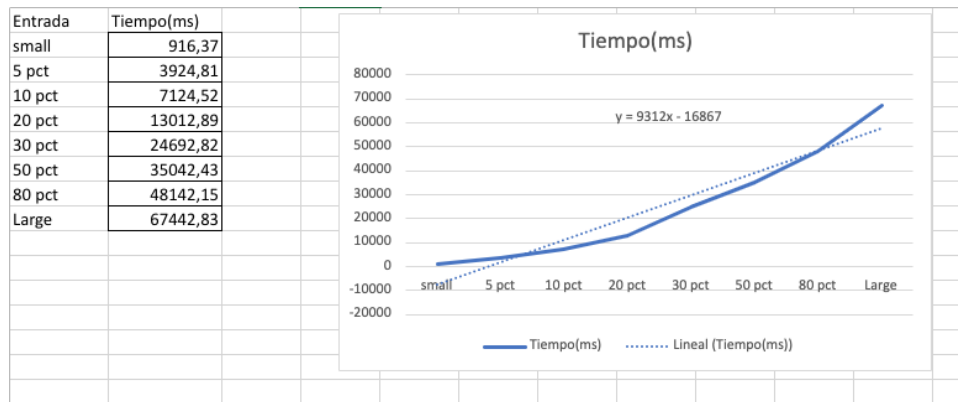
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	916.37
5 pct	Dato2	3924.81
10 pct	Dato3	7124.52
20 pct	Dato4	13012.89
30 pct	Dato5	24692.82
50 pct	Dato6	35042.43
80 pct	Dato7	48142.15
large	Dato8	67442.83

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

Análisis de resultados de la implementación, tener cuenta las pruebas realizadas y el analisis de complejidad.

Requerimiento <<6>>

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

Descripción

Este requerimiento no tiene parámetros de entrada. Lo primero que hace es aplicar el algoritmo de kosaraju. Así se pueden obtener el número de manadas. Después se obtiene la estructura de datos, donde indica a que manada pertenece cada nodo. Después se hace un mapa con todos los nodos de cada manada y después se verifica cuáles son las primeras 5 manadas, es decir, las que cuentan con más nodos. Finalmente, se retornan las primeras 5 manadas con su respectiva información

Entrada	Fecha inicial del análisis (con formato "%Y-%m-%d"). <ul style="list-style-type: none">Fecha final del análisis (con formato "%Y-%m-%d").El sexo registrado del animal (animal-sex).
Salidas	El individuo que recorrió más distancia dentro en el rango La ruta más larga del individuo registrada dentro del rango El individuo que recorrió menos distancia dentro en el rango

	La ruta más larga del individuo registrada del rango
Implementado (Sí/No)	Si, Mateo Rincón, Mateo Restrepo y Juan Puig

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Obtener lista de lobos	$O(1)$
Revisar cada lobo	$O(N)$
Reconstruir camino del lobo y sumar distancias	$O(N)$
Comprobar si es el mayor	$O(1)$
Repetir	$O(N^2)$
TOTAL	$O(2n^2)$

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).

Procesadores	Intel core i5
Memoria RAM	8 GB
Sistema Operativo	MacOS

Entrada	Tiempo (s)
small	1896.35
5 pct	6274.14
10 pct	13524.93
20 pct	23012.89
30 pct	32632.72
50 pct	55012.32
80 pct	75942.32
large	92482.92

Tablas de datos

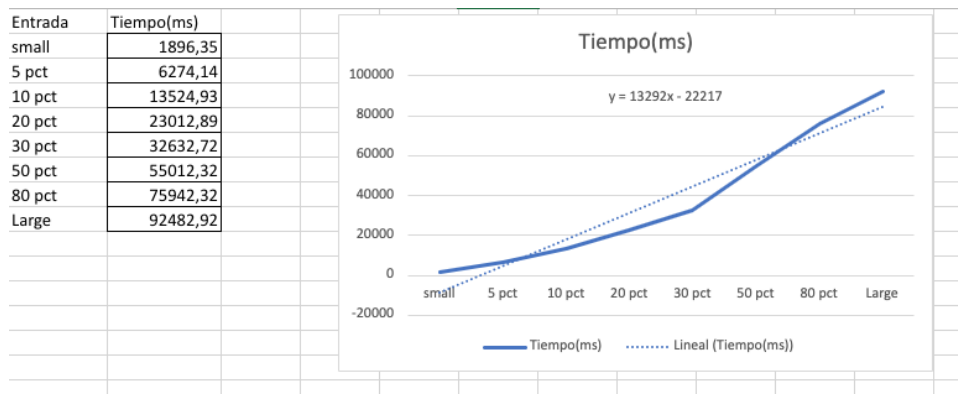
Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	1896.35

5 pct	Dato2	6274.14
10 pct	Dato3	13524.93
20 pct	Dato4	23012.89
30 pct	Dato5	32632.72
50 pct	Dato6	55012.32
80 pct	Dato7	75942.32
large	Dato8	92482.92

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

El requerimiento 6 muestra una gráfica con un comportamiento exponencial. Cómo era predecible, a medida que se necesita trabajar con más datos también se usa más tiempo al obtener la respuesta del requerimiento. Por esto es que siempre la gráfica tiene una pendiente positiva. Además, un punto notable de la gráfica obtenida es que forma una ligera curva que cada vez es más y más pronunciada. Esto sucede debido a la complejidad del requerimiento de $O(2N^2)$. Finalmente, es posible concluir que la gráfica refleja en cierta manera los resultados teóricos esperados del análisis de complejidad.

Requerimiento <<7>>

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

Descripción

Este requerimiento no tiene parámetros de entrada. Lo primero que hace es aplicar el algoritmo de kosaraju. Así se pueden obtener el número de manadas. Después se obtiene la estructura de datos, donde indica a que manada pertenece cada nodo. Después se hace un mapa con todos los nodos de cada manada y después se verifica cuáles son las primeras 5 manadas, es decir, las que cuentan con más nodos. Finalmente, se retornan las primeras 5 manadas con su respectiva información

Entrada	Fecha inicial del análisis (con formato "%Y-%m-%d"). <ul style="list-style-type: none">• Fecha final del análisis (con formato "%Y-%m-%d").• Temperatura ambiente mínima (en grados centígrados).• Temperatura ambiente máxima (en grados centígrados).
Salidas	El total de manadas reconocidas por sus movimientos y puntos de encuentro (componentes conectados) en el rango de fechas y temperatura ambiente dados. <ul style="list-style-type: none">• Mostrar los tres primeros y tres últimas manadas con mayor dominio sobre el territorio (de mayor a menor número de puntos de encuentro dentro del componente conectado) La ruta más larga posible dentro del territorio identificado (componente fuertemente conectado)
Implementado (Sí/No)	Sí, Mateo Rincón, Mateo Restrepo y Juan Puig

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Filtrar datos	$O(N)$
Nuevo grafo	$O(N)$
Kosaraju	$O(N \log(N))$
Comprobar manadas	$O(6)$
TOTAL	$O(N^3 \log N)$

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).

Procesadores	Intel core i5
Memoria RAM	8 GB
Sistema Operativo	MacOS

Entrada	Tiempo (s)
small	716.37
5 pct	3214.48
10 pct	7164.26
20 pct	12313.54
30 pct	28999.71
50 pct	43847.88
80 pct	69747.16
large	77428.73

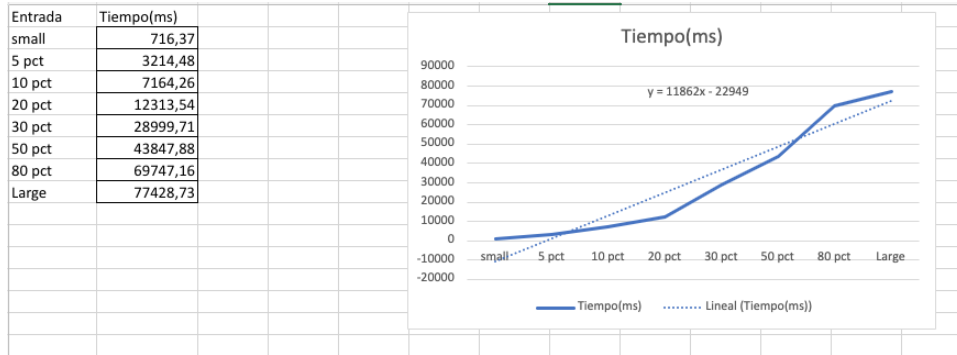
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	716.37
5 pct	Dato2	3214.48
10 pct	Dato3	7164.26
20 pct	Dato4	12313.54
30 pct	Dato5	28999.71
50 pct	Dato6	43847.88
80 pct	Dato7	69747.16
large	Dato8	77428.73

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

El requerimiento 7 es el último y el más complejo de todos los requerimientos. Esto queda evidenciado en la gráfica. De entrada, los tiempos de ejecución son los más elevados, aunque no es nada sorprendente ya que para el desarrollo de este requerimiento es necesario un nuevo grafo, lo que hace más demorado el proceso, aparte de tener que filtrar una lista de más de 300 mil elementos. Esto evidentemente, se va haciendo más demorado a medida que aumentan los datos que se están ejecutando. Por otro lado, la curva característica de un crecimiento exponencial es más evidente que en cualquier otro requerimiento, lo que muestra congruencia con su complejidad teórica de $O(N^3 \log N)$. A lo último existe una anomalía ya que la curva parece disminuir un poco pero creemos que de repetir una y otra vez las pruebas sería más evidente el crecimiento, aparte, con archivos mucho más grandes también se podría ver mejor el crecimiento.

Requerimiento Ejemplo

Descripción

```
def get_data(data_structs, id):
    """
    Retorna un dato a partir de su ID
    """
    pos_data = lt.isPresent(data_structs["data"], id)
    if pos_data > 0:
        data = lt.getElement(data_structs["data"], pos_data)
        return data
    return None
```

Este requerimiento se encarga de retornar un dato de una lista dado su ID. Lo primero que hace es verificar si el elemento existe. Dado el caso que exista, retorna su posición, lo busca en la lista y lo retorna. De lo contrario, retorna None.

Entrada	Estructuras de datos del modelo, ID.
Salidas	El elemento con el ID dado, si no existe se retorna None

Implementado (Sí/No)	Si. Implementado por Juan Andrés Ariza
----------------------	--

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Buscar si el elemento existe (isPresent)	O(n)
Obtener el elemento (getElement)	O(1)
TOTAL	O(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	8 GB
Sistema Operativo	Windows 10

Entrada	Tiempo (ms)
small	0.05
5 pct	0.33
10 pct	1.28
20 pct	2.54
30 pct	4.98
50 pct	7.51
80 pct	13.81
large	25.97

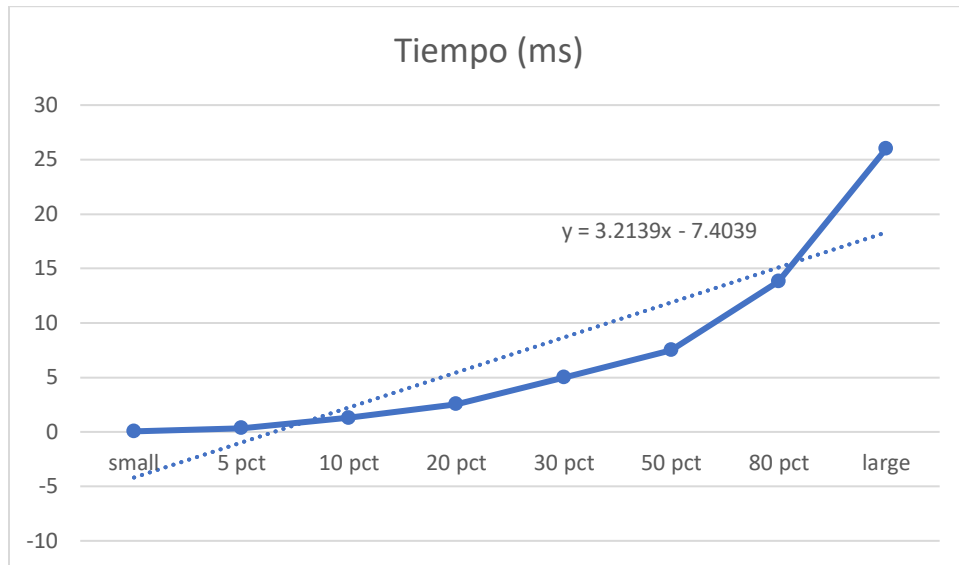
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	0.05
5 pct	Dato2	0.33
10 pct	Dato3	1.28
20 pct	Dato4	2.54
30 pct	Dato5	4.98
50 pct	Dato6	7.51
80 pct	Dato7	13.81
large	Dato8	25.97

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

A pesar de que obtener un elemento en un *ArrayList*, dada su posición, tiene complejidad constante, la implementación de este requerimiento tiene un orden lineal $O(n)$. Esto debido a que, lo primero que se hace es verificar si el elemento hace parte de la lista. Específicamente, a la hora de buscar un elemento en una lista, en el peor de los casos es necesario recorrer toda la lista, es decir, complejidad lineal.

Este comportamiento se puede evidenciar experimentalmente en la gráfica. Ya que, gracias a que los datos no se encuentran tan dispersos con respecto a la línea de tendencia, la curva coincide con el comportamiento lineal esperado.