

Análisis de Resultados

Reto 3 Grupo 4

Integrantes

Ashlee Yin Romero, a.yin@uniandes.edu.co 202421132

Daniel Galindo, d.galindot@uniandes.edu.co, 202414673

Tomás Lozano, t.lozanoc@uniandes.edu.co 202422837

Especificaciones

La medición del tiempo de las funciones fueron tomadas en un computador MacBook Air 2015, viejita pero bien cuidada, con un procesador 1,6 GHz Dual-Core Intel Core i5 y memoria de 8 GB 1600 MHz DDR3. Todos los tiempos fueron tomados en milisegundos.

Carga de Datos

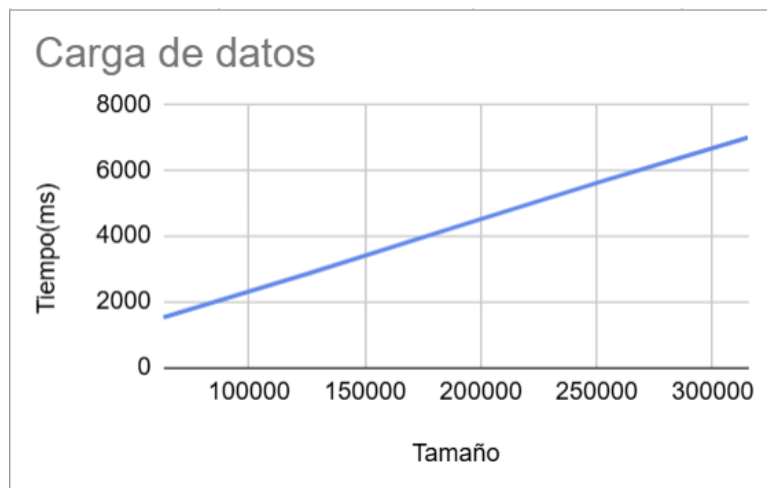
Descripción: Cargar y almacenar los datos en estructuras eficientes.

Complejidad: $O(n \log n)$ (por uso de RBT y ordenamientos).

Análisis:

Durante la carga, los datos son insertados en listas, mapas hash y árboles RBT para facilitar búsquedas eficientes. Las fechas se convierten a objetos datetime, y se organizan en estructuras categorizadas por fecha de ocurrido, fecha de reporte, edad, y área. Esta organización permite el uso posterior de búsquedas por rango con complejidades controladas. Aunque la inserción en listas y mapas tiene costo lineal, el ordenamiento aplicado posteriormente implica un costo dominante de $O(n \log n)$.

	CARGA DE DATOS	
Archivo	Tamaño	Tiempo(ms)
Crime_in_LA_20	63088	1534.908
Crime_in_LA_40	126176	2876.294
Crime_in_LA_60	189264	4280.275
Crime_in_LA_80	252352	5666.278
Crime_in_LA_100	315440	6994.953



Requerimiento 1 (Grupal)

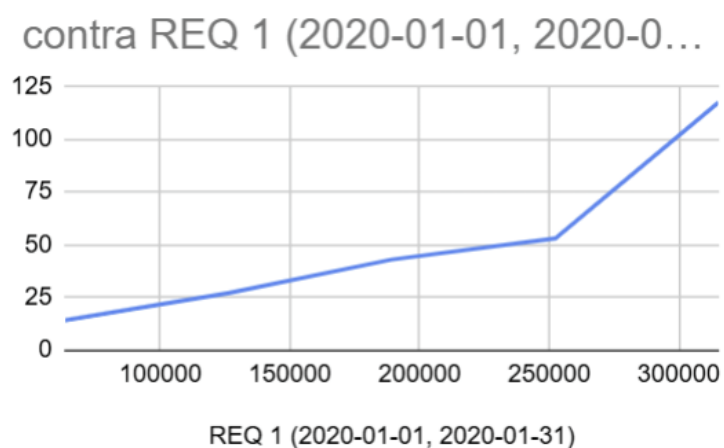
Descripción: Listar los crímenes ocurridos entre dos fechas.

Complejidad: $O(n)$ (recorrido lineal de los nodos del árbol en el rango)

Análisis:

El requerimiento se ejecuta en tiempo lineal respecto al número de registros recorridos, ya que se hace un recorrido por el árbol binario balanceado (RBT) entre las fechas dadas. El uso de estructuras eficientes permite mantener un buen rendimiento.

REQ 1 (2020-01-01, 2020-01-31)		
Archivo	Tamaño	Tiempo(ms)
Crime_in_LA_20	63088	14.23269999
Crime_in_LA_40	126176	27.0957
Crime_in_LA_60	189264	42.93640003
Crime_in_LA_80	252352	52.8743
Crime_in_LA_100	315440	117.3022



Requerimiento 2 (Grupal)

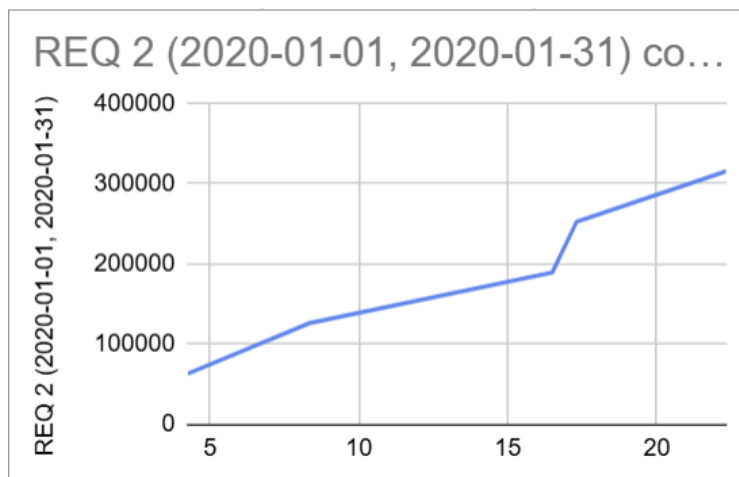
Descripción: Listar los crímenes graves resueltos reportados en un rango de fechas.

Complejidad: $O(n \log n)$ (filtrado + ordenamiento por fecha y área)

Análisis:

Aunque se recorre el árbol como en el req 1, se aplica un ordenamiento adicional descendente por fecha y suborden por área, lo que agrega una complejidad logarítmica adicional. Se aplica slicing para mostrar los 5 primeros y 5 últimos.

REQ 2 (2020-01-01, 2020-01-31)		
Archivo	Tamaño	Tiempo(ms)
Crime_in_LA_20	63088	4.245400012
Crime_in_LA_40	126176	8.340499997
Crime_in_LA_60	189264	16.4885
Crime_in_LA_80	252352	17.30880001
Crime_in_LA_100	315440	22.34120002



Requerimiento 3 (Individual)

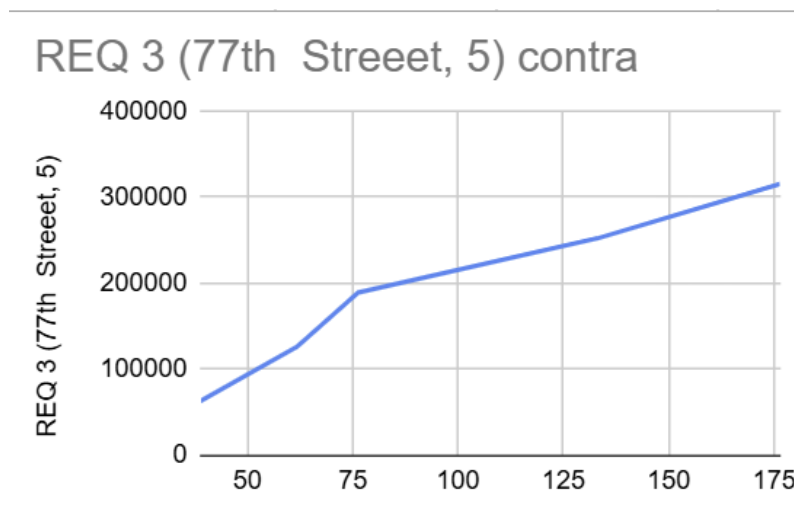
Descripción: Consultar los N crímenes reportados más recientemente para un área.

Complejidad: $O(n \log n)$ (heap de máximos por fecha de reporte)

Análisis:

Se filtran los registros por área desde un mapa hash y se construye un heap con clave inversa para obtener los N crímenes más recientes. El ordenamiento es eficiente para este caso.

REQ 3 (77th Street, 5)		
Archivo	Tamaño	Tiempo(ms)
Crime_in_LA_20	63088	38.87
Crime_in_LA_40	126176	61.74
Crime_in_LA_60	189264	76.32
Crime_in_LA_80	252352	133.29
Crime_in_LA_100	315440	176.49



Requerimiento 4 (Individual)

Descripción: Consultar los N crímenes con víctimas en un rango de edad y clasificarlos por gravedad.

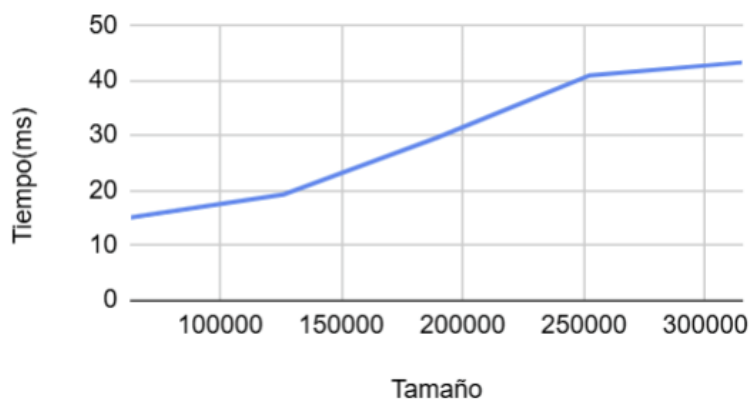
Complejidad: $O(n \log n)$

Análisis:

Se hace una búsqueda por rango de edad en el árbol por edades. Luego, se clasifican por gravedad y se ordenan por edad y fecha. El procedimiento cumple con las recomendaciones del enunciado.

REQ 4 (50,70,10)		
Archivo	Tamaño	Tiempo(ms)
Crime_in_LA_20	63088	15.03380001
Crime_in_LA_40	126176	19.23180002
Crime_in_LA_60	189264	29.50939998
Crime_in_LA_80	252352	40.8547
Crime_in_LA_100	315440	43.22490001

REQ 4 (50,70,10)



Requerimiento 5 (Individual)

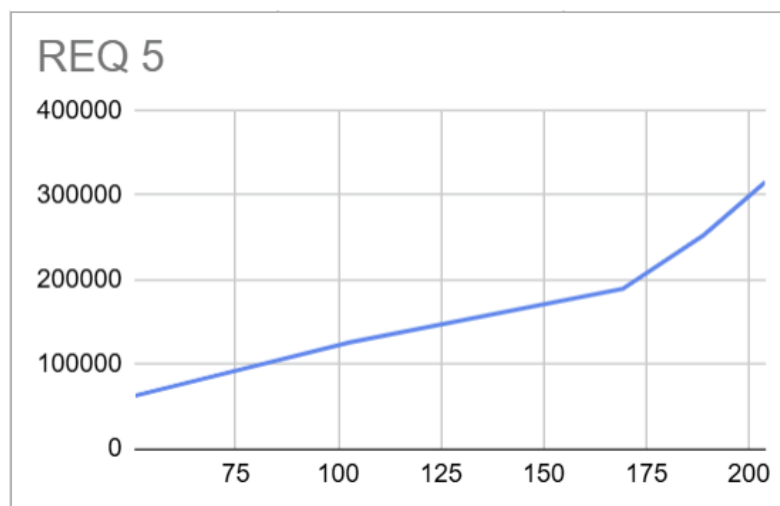
Descripción: Consultar las N áreas con mayor cantidad de crímenes no resueltos en un rango de fechas.

Complejidad: $O(n \log n)$ (agrupación + heap por cantidad)

Análisis:

Se recorren los crímenes por fecha y se agrupan por área con un contador. Luego, se aplica un heap para determinar las áreas con mayor número de casos.

REQ 5		
Archivo	Tamaño	Tiempo(ms)
Crime_in_LA_20	63088	50.4461
Crime_in_LA_40	126176	102.9008
Crime_in_LA_60	189264	169.2851
Crime_in_LA_80	252352	188.8963
Crime_in_LA_100	315440	204.0504999



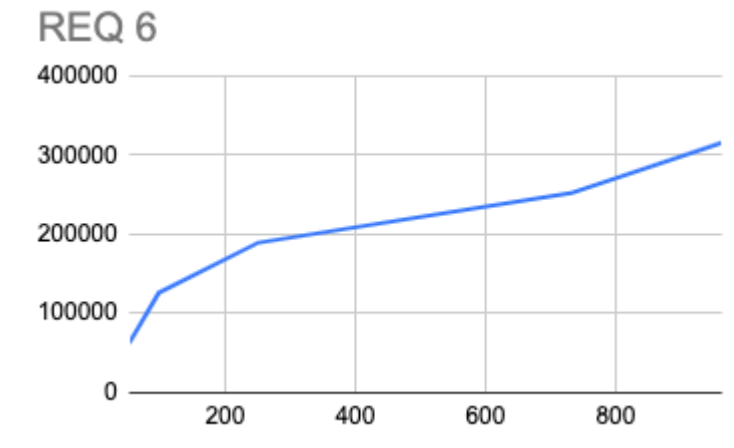
Requerimiento 6

Descripción: Consultar las N áreas más seguras para un sexo, en un mes específico del año.

Complejidad: $O(n \log n)$ (acceso + heap por cantidad)

Análisis:

Se accede a la tabla de hash por areas y se calcula para cada una de ellas la cantidad de crímenes en el mes. Luego con un heap se sacan los menores n.



REQ 6 (n=5, F, mes 3)		
Archivo	Tamaño	Tiempo(ms)
Crime_in_LA_20	63088	53.078067
Crime_in_LA_40	126176	98.036232
Crime_in_LA_60	189264	250.36
Crime_in_LA_80	252352	732.81
Crime_in_LA_100	315440	962.73

Requerimiento 7 (Grupal)

Descripción: Determinar los N crímenes más comunes para víctimas de un sexo en un rango de edad.

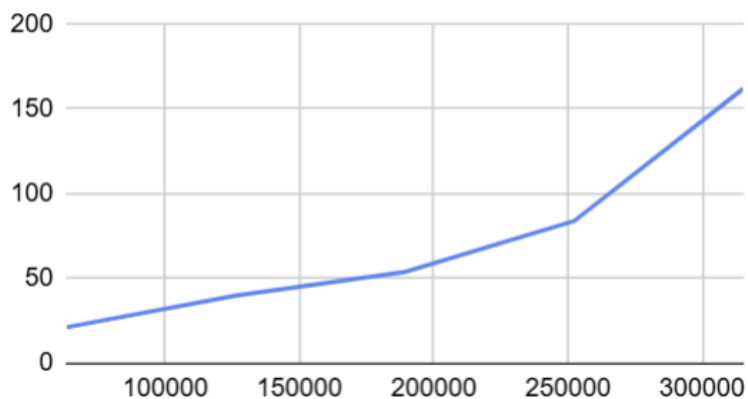
Complejidad: $O(n \log n)$ (filtrado + conteo + ordenamiento)

Análisis:

Se realiza una búsqueda en el árbol por edad, se filtran por sexo y se agrupan los delitos por código. Luego se cuentan por edad y por año para cada crimen. Los datos se presentan como lo exige el requerimiento.

REQ 7 (5,F,20,30)		
Archivo	Tamaño	Tiempo(ms)
Crime_in_LA_20	63088	21.22040001
Crime_in_LA_40	126176	39.85569999
Crime_in_LA_60	189264	53.76199999
Crime_in_LA_80	252352	83.87529999
Crime_in_LA_100	315440	161.9265

REQ 7 (5,F,20,30)



Requerimiento 8 (BONO)

REQ 8 n=5, 510, Topanga		
Archivo	Tamaño	Tiempo(ms)
Crime_in_LA_20	63088	20659.95
Crime_in_LA_40	126176	90093.53
Crime_in_LA_60	189264	267780.23
Crime_in_LA_80	252352	415634.04
Crime_in_LA_100	315440	512130.43

