

## Documento de análisis reto 1

Alejandro Borda – 202020727

Juan Sebastián Lache - 201821331

Máquina de prueba:

Procesadores => AMD Ryzen 5 3550H 2.10 GHz

Memoria RAM => 8 GB

Sistema operativo => Windows 10, 64 Bits

Carga de datos:

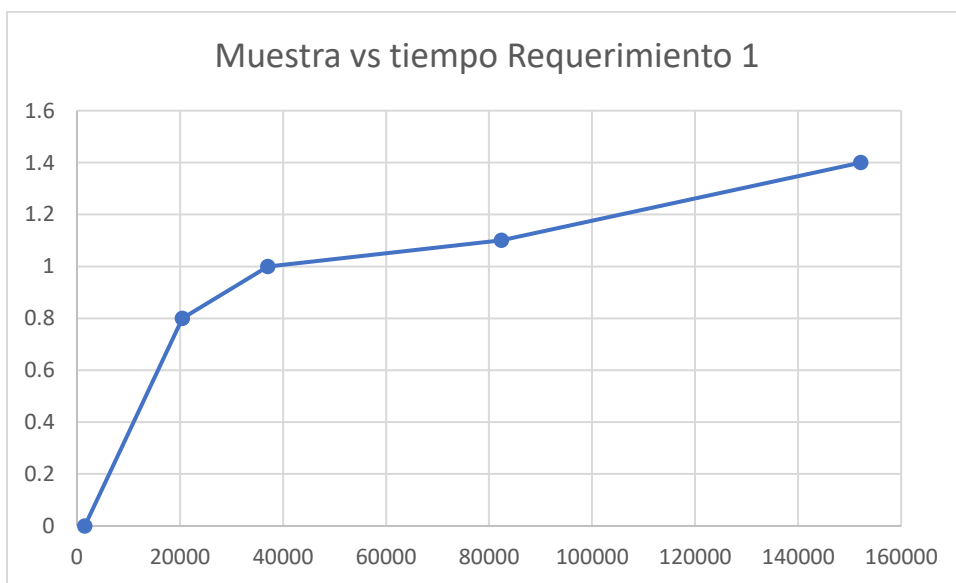
Muestra	Tiempo (segundos)
1523 (-small)	1
20471 (10%)	88
37020 (20%)	180
82376 (50%)	510
152180 (-large)	1070



En base a los resultados obtenidos para las diferentes pruebas y la grafica de tendencia, además de la información otorgada por el código de la carga de datos consideramos que el crecimiento temporal de la carga de datos del programa es de  $O(n)$ .

#### Requerimiento 1

Muestra	Tiempo (segundos)
1523 (-small)	0
20471 (10%)	0.8
37020 (20%)	1
82376 (50%)	1.1
152180 (-large)	1.4

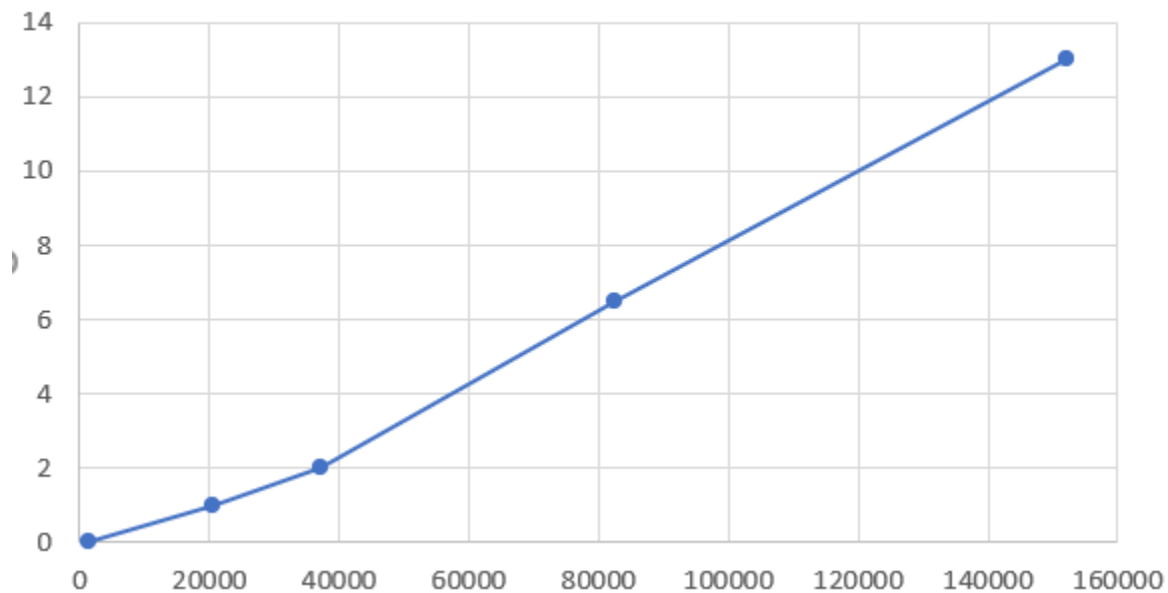


En base a la grafica se podría creer que el crecimiento temporal es de tipo  $O(\log n)$ , sin embargo esto se puede deber a la poca variación entre las mediciones del tiempo y en base al código es mas posible que su crecimiento sea de tipo  $O(n \log n)$  debido al uso de merge sort.

#### Requerimiento 2

Muestra	Tiempo (segundos)
1523 (-small)	0
20471 (10%)	1
37020 (20%)	2
82376 (50%)	6.5
152180 (-large)	13

## REQ 2, tiempo vs muestra



En base a los resultados obtenidos para las diferentes pruebas y la grafica de tendencia, además de la información otorgada por el código del requerimiento 2 consideramos que el crecimiento temporal de la carga de datos del programa es de  $O(n \log n)$ . Tambien es posible especular que crece de tipo  $O(n \log n)$  debido a que la lista se ordena por medio de merge sort.

Requerimiento 3 (Juan Sebastian Lache):

\*Para esta prueba se usó una máquina diferente:

Procesador: 2,6 GHz Intel Core i5

Memoria RAM: 8 GB

Sistema operativo: MacOS Mojave

Muestra	Tiempo (segundos)
1523 (-small)	0.354
20471 (10%)	67.947
37020 (20%)	203.58
82376 (50%)	706.59
152180 (-large)	1653.28



Con base en los Resultados obtenidos en las pruebas y con la gráfica, considero que la complejidad temporal del algoritmo es de  $O(n^3)$ , esta complejidad se adecua al ver el código de esta prueba.

Requerimiento 4 (Alejandro Borda):

Muestra	Tiempo (segundos)
1523 (-small)	0.5
20471 (10%)	1
37020 (20%)	1.1
82376 (50%)	1.3
152180 (-large)	1.5



En base a los resultados de la tabla y la gráfica realizada, se podría considerar que la complejidad temporal del algoritmo es de  $O(\log n)$ , sin embargo esto no es intuitivo al ver el código y debido a el poco tiempo que se demora en ejecutar y las pequeñas diferencias entre pruebas es mas complicado afirmar con seguridad la tendencia que se presenta, sin embargo es claro que no se trata de una constante, pues se evidencia una relación entre el numero de datos y el tiempo de ejecución.

Requerimiento 5:

Muestra	Tiempo (segundos)
1523 (-small)	0
20471 (10%)	1.2
37020 (20%)	2
82376 (50%)	4.5
152180 (-large)	9

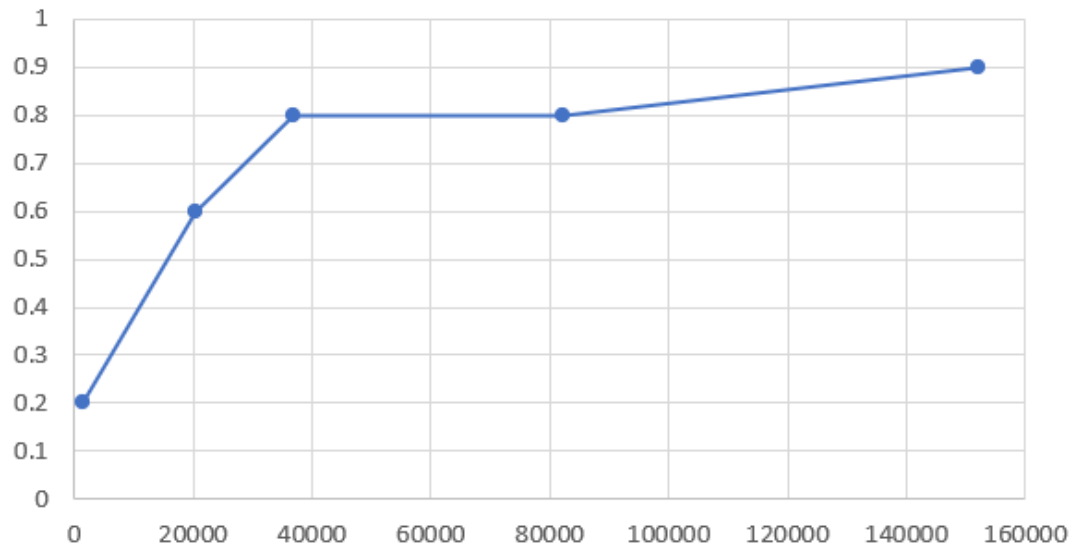


En base al crecimiento de los datos y el tiempo es valido presumir que su crecimiento temporal es  $O(n)$ , esto es apoyado claramente por la grafica y es razonable en base al código implementado.

Requerimiento 6:

Muestra	Tiempo (segundos)
1523 (-small)	0.2
20471 (10%)	0.6
37020 (20%)	0.8
82376 (50%)	0.8
152180 (-large)	0.9

REQ 6, tiempo vs muestra



Nuevamente no se nota una complejidad intuitiva en el código y la poca variación de tiempo entre muestras y el pequeño tiempo que se demora en ejecutar el algoritmo hace un poco complejo el análisis, sin embargo es claro que la complejidad temporal es dependiente de la muestra lo cual implica que no es una constante y en base a la grafica parece ser posible que la complejidad fuera  $O(\log n)$ , sin embargo por las complicaciones previamente mencionadas no se descarta que sea de tipo  $O(n)$  a pesar de no parecerlo a primera instancia.

