

Nicolas Camargo (202020782)

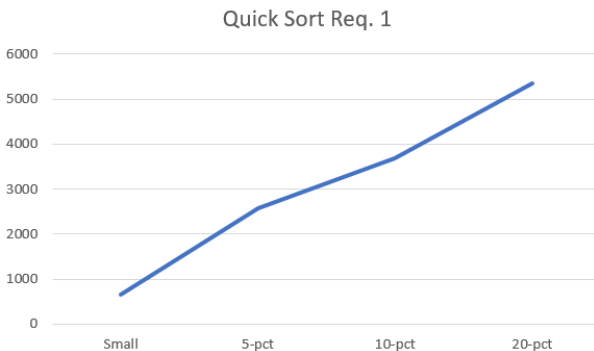
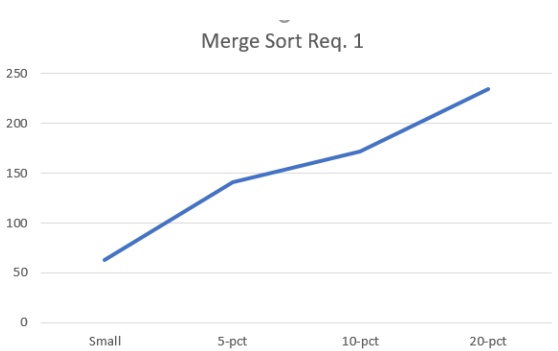
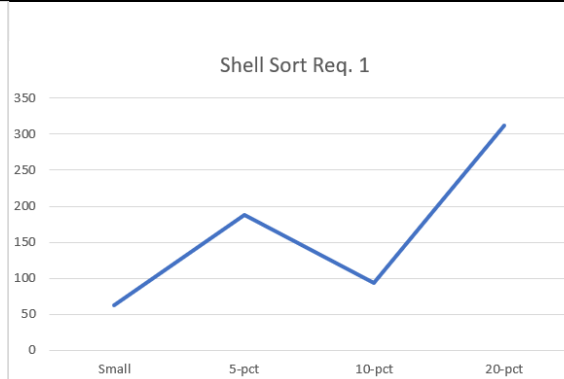
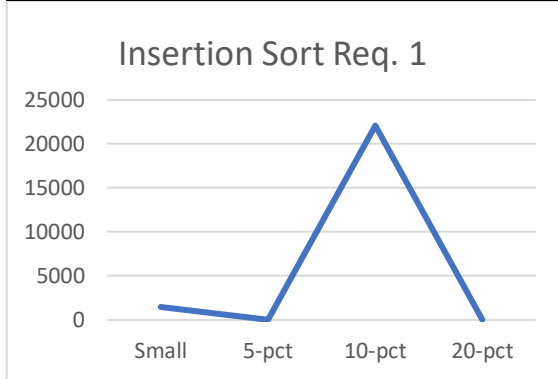
Juan Manuel Pérez Sanchez (202021827)

## Análisis de Complejidad Reto 1

### Requerimiento 1.

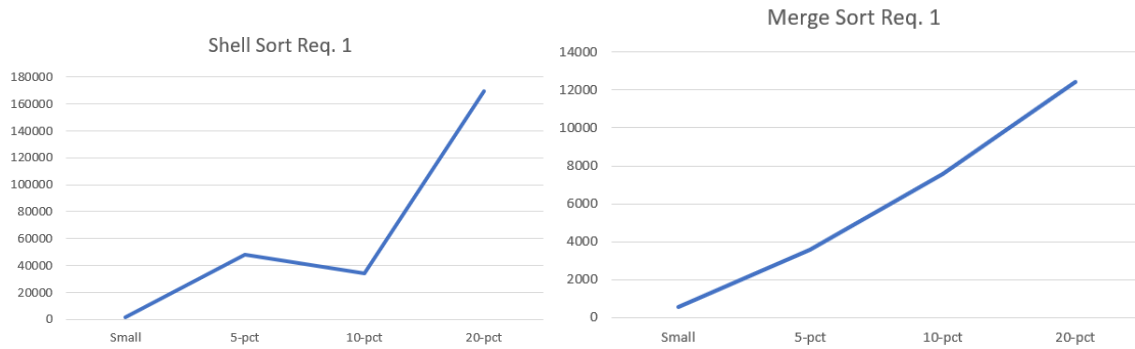
*Arreglo*

Datos	Insertion	Shell	Merge	Quick Sort
Small	1437.50	62.50	62.50	656.25
5-pct	15.625	187.5	140.625	2562.5
10-pct	22078.125	93.75	171.875	3687.5
20-pct	31.25	312.5	234.375	5343.75



*Lista Encadenada*

Datos	Insertion	Shell	Merge	Quick Sort
Small	234.375	1296.875	578.125	132781.25
5-pct	-	48156.25	3562.5	-
10-pct	-	33937.5	7562.5	-
20-pct	-	169468.75	12406.25	-

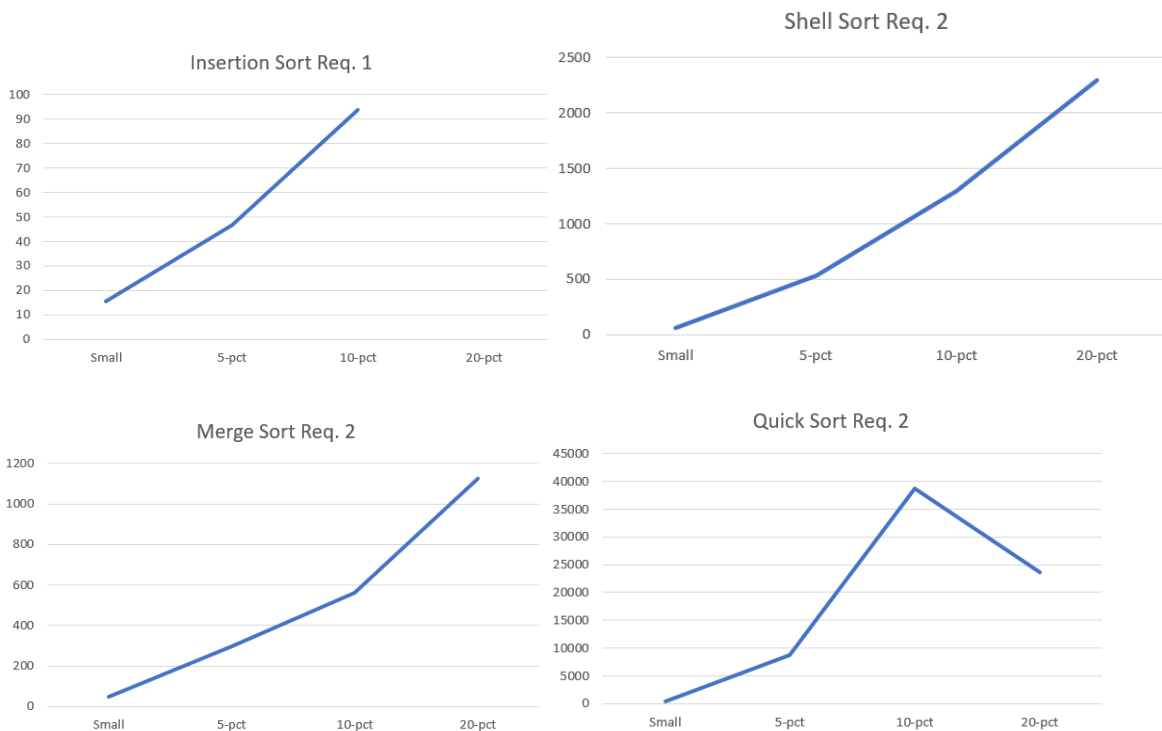


Después de ver los tiempos de ejecución se decidió utilizar la estructura de datos para arreglos. Analizando el mejor tiempo de ejecución se ve en el ordenamiento de Merge. Según el algoritmo la complejidad es de los while de  $O(n^2)$ , y el Merge tiene un promedio de  $O(n \log(n))$ . Haciendo el respectivo calculo se tiene una complejidad de  $O(n^3 \log(n))$ .

## Requerimiento 2.

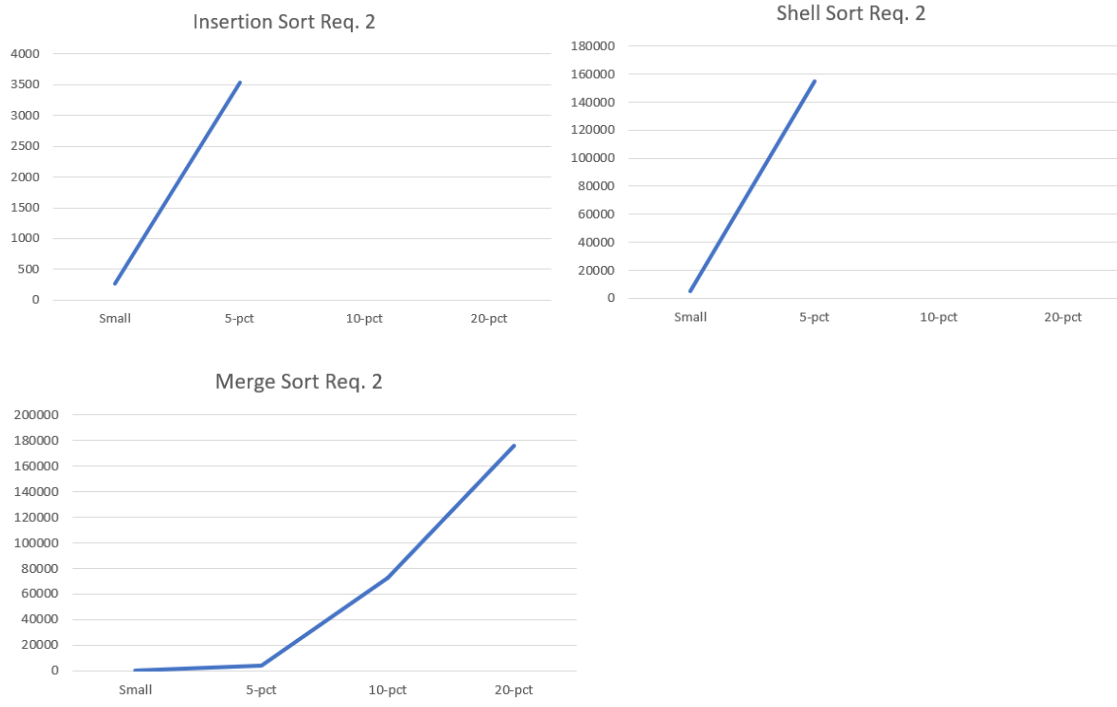
### Arreglo

Datos	Insertion	Shell	Merge	Quick Sort
Small	15.625	62.5	46.875	421.875
5-pct	46.875	531.25	296.875	8781.25
10-pct	93.75	1296.875	562.5	38734.375
20-pct	-	2296.875	1125.0	23609.375



### Lista Encadenada

Datos	Insertion	Shell	Merge	Quick Sort
Small	265.625	5187.5	562.5	121765.625
5-pct	3531.25	154828.125	4125.0	-
10-pct	-	-	72843.75	-
20-pct	-	-	175796.875	-



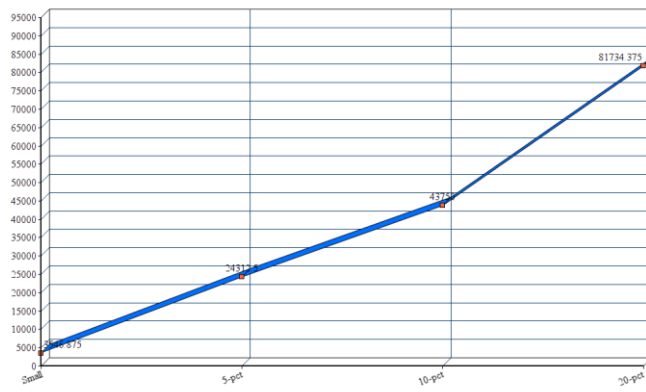
Después de ver los tiempos de ejecución se decidió implementar los algoritmos de arreglo de ejecución. El menor tiempo se da en el ordenamiento de Merge, sin embargo, cabe resaltar que el ordenamiento por Insertion es muy irregular ya que hay momentos en los que parece ser el más efectivo y luego hay un aumento significativo en el tiempo de ejecución. La complejidad de este algoritmo es de de dos ciclos while con una complejidad de  $O(n)$  cada uno, el Merge que tiene  $O(n \log(n))$ . Haciendo los cálculos la complejidad de este algoritmo es de  $O(n^3 \log(n))$ .

### Requerimiento 3. (Nicolas Camargo)

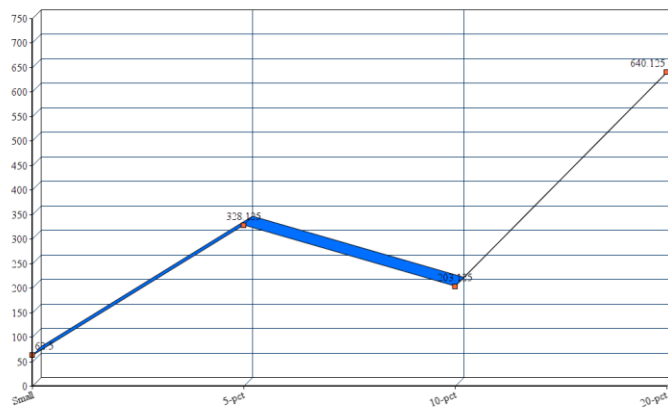
#### Arreglo

Datos	Insertion	Shell	Merge	Quick Sort
Small	3546.875	62.5	140.625	-
5-pct	24312.5	328.125	343.75	-
10-pct	43750.0	203.125	343.75	-
20-pct	81734.375	640.125	437.5	-

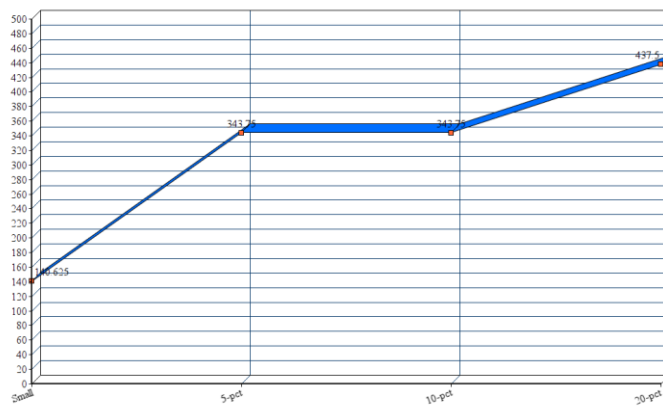
Insertion sort



Shell sort



Merge Sort

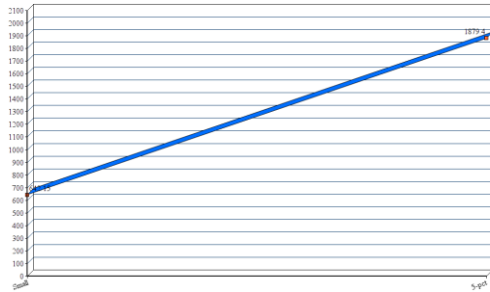


## Lista Encadenada

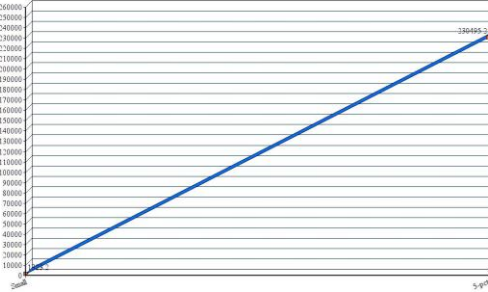
Datos	Insertion	Shell	Merge	Quick Sort
-------	-----------	-------	-------	------------

Small	642.15	1323.2	748.9	-
5-pct	1879.4	230495.2	1467.875	-
10-pct	-	-	-	-
20-pct	-	-	-	-

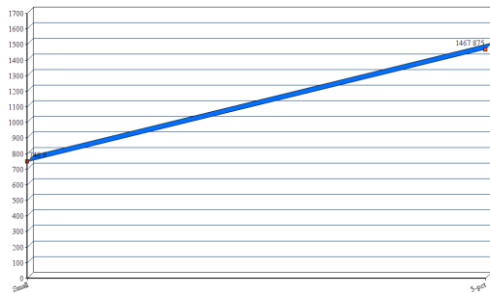
Insertion sort



Shell sort



Merge sort



Como se puede observar en los resultados obtenidos, el uso del arreglo para este requerimiento tuvo mejores resultados para el uso de Quick sort y para 10pct y 20pct. Sin embargo, tuvo peores tiempos que los que se observan en el uso de la lista encadenada, pues esta tuvo buenos tiempos para todos los sorts pero solo en los conjuntos de datos más pequeños. Se observa una complejidad logarítmica  $O(\log n)$  para insertion sort en el arreglo, mientras que el resto conservan una complejidad de  $O(n)$  debido a los ciclos usados.

#### Requerimiento 4. (Juan Manuel Pérez)

##### Arreglo

Datos	Insertion	Shell	Merge	Quick Sort
Small	0.0	0.0	0.0	0.0
5-pct	0.0	0.0	0.0	0.0
10-pct	0.0	0.0	0.0	0.0
20-pct	0.0	0.0	0.0	15.625

##### Lista Encadenada

Datos	Insertion	Shell	Merge	Quick Sort
Small	0.0	0.0	0.0	0.0
5-pct	0.0	0.0	0.0	0.0

10-pct	0.0	0.0	0.0	0.0
20-pct	0.0	0.0	0.0	15.625

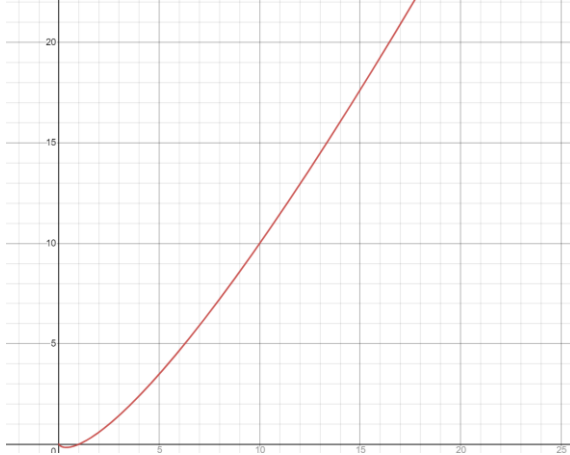


Gráfico  $O(n \log (n))$

Este algoritmo tiene tiempos de ejecución bajos, a comparación de los otros requerimientos es el que menor tiempo de ejecución tiene. Dado el patrón de los otros requerimientos y la complejidad del ordenamiento Merge siendo  $o(n \log (n))$  en el peor de los casos consideramos que este ordenamiento es el mas efectivo. La complejidad de este algoritmo es de  $o(n \log (n))$  ya que solo cuenta con el ordenamiento Merge.

## Requerimiento 5.

### Arreglo

Datos	Insertion	Shell	Merge	Quick Sort
Small	1250.0	125.0	78.125	125.0
5-pct	117593.75	515.125	734.375	1773.125
10-pct	649906.25	765.625	1406.25	44718.75
20-pct				

### Lista Encadenada

Datos	Insertion	Shell	Merge	Quick Sort
Small				
5-pct				
10-pct				
20-pct				