Nicolas Camargo (202020782)

Juan Manuel Pérez Sanchez (202021827)

Análisis de Complejidad Reto 2

Requerimiento 1.

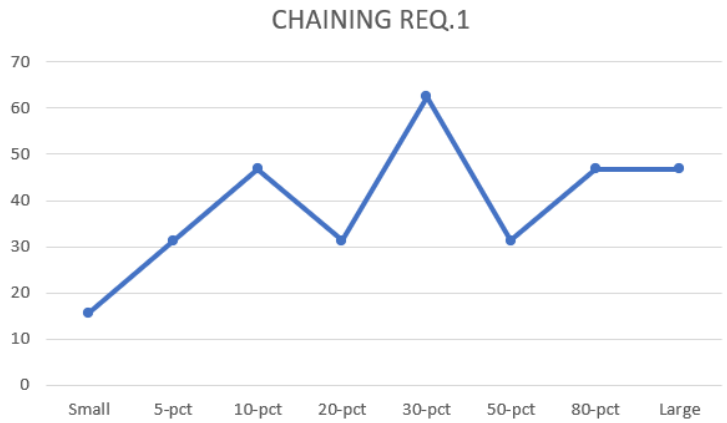
Tabla de Comparación de complejidad

	RETO 1	RETO 2
Complejidad	O(n log (n))	O(n log (n))

Tiempos de ejecución

CHAINING

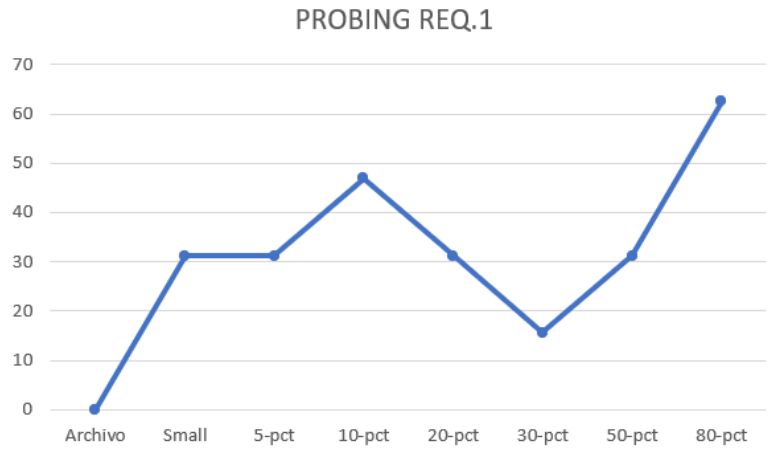
Archivo	Tiempo (mseg)
Small	15.625
5-pct	31.25
10-pct	46.875
20-pct	31.25
30-pct	62.5
50-pct	31.25
80-pct	46.875
Large	46.875



PROBING

Archivo	Tiempo (mseg)
Small	31.25
5-pct	31.25
10-pct	46.875
20-pct	31.25

30-pct	15.625
50-pct	31.25
80-pct	62.5
Large	46.875



Para el Requerimiento 1 la complejidad de este algoritmo se da en el merge sort que tiene una complejidad de O(n log (n)). Dado los tiempos de ejecución son muy similares por lo que se escogió Linear Probing ya que utiliza menos memoria.

Requerimiento 2.

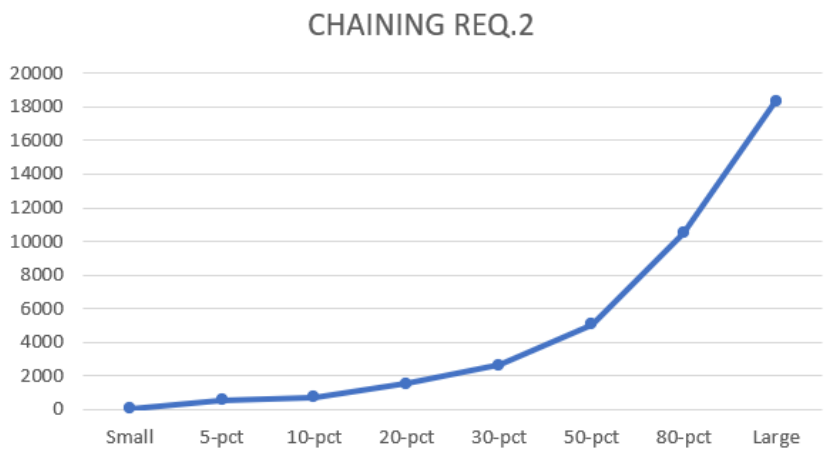
Tabla de Comparación de complejidad

	RETO 1	RETO 2
Complejidad	O(n log (n))	O(n log (n))

Tiempos de ejecución

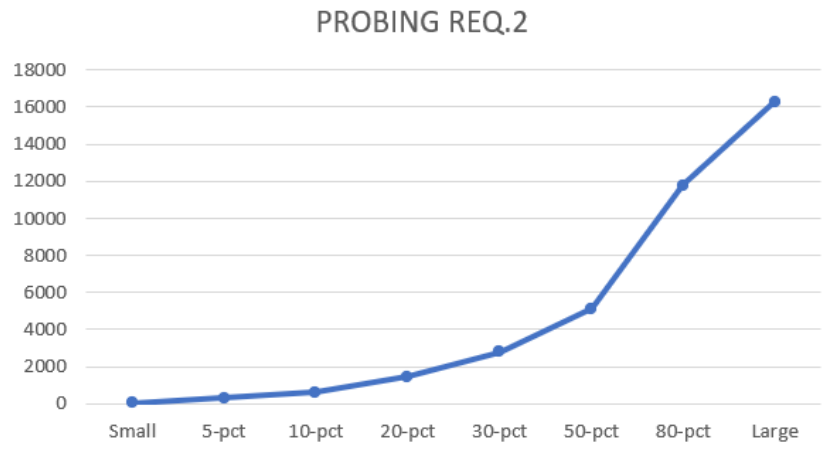
CHAINING

Archivo	Tiempo (mseg)
Small	62.5
5-pct	546.875
10-pct	734.375
20-pct	1515.625
30-pct	2656.25
50-pct	5078.125
80-pct	10500.0
Large	18359.375



PROBING

Archivo	Tiempo (mseg)
Small	31.25
5-pct	328.125
10-pct	625.0
20-pct	1453.125
30-pct	2781.25
50-pct	5109.375
80-pct	11781.25
Large	16281.25



El Requerimiento 2 la complejidad de este algoritmo también se da en el merge sort con una complejidad de O(n log (n)). Los tiempos de ejecución son un poco mas rápidos en Linear Probing por lo que se escogió.

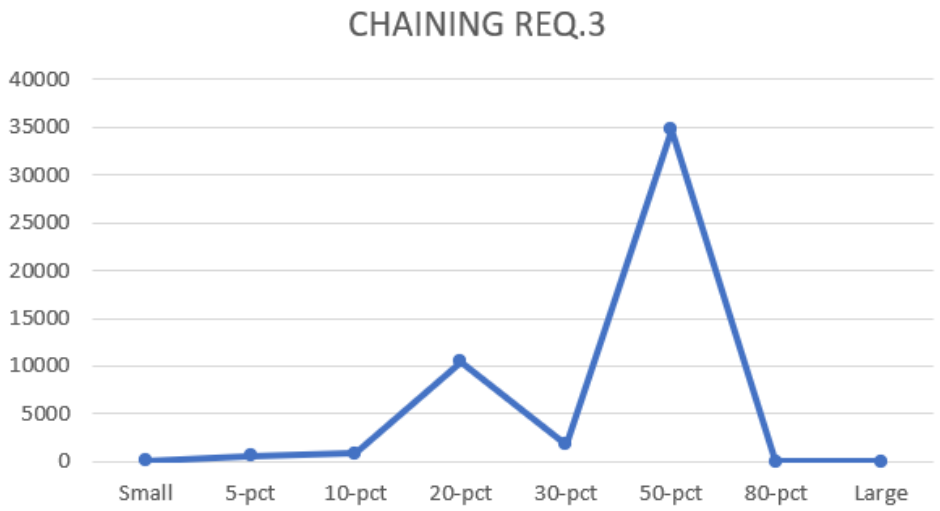
Requerimiento 3. (Nicolas Camargo)

Tabla de Comparación de complejidad

	RETO 1	RETO 2
Complejidad	O(n log(n))	O(n log (n))

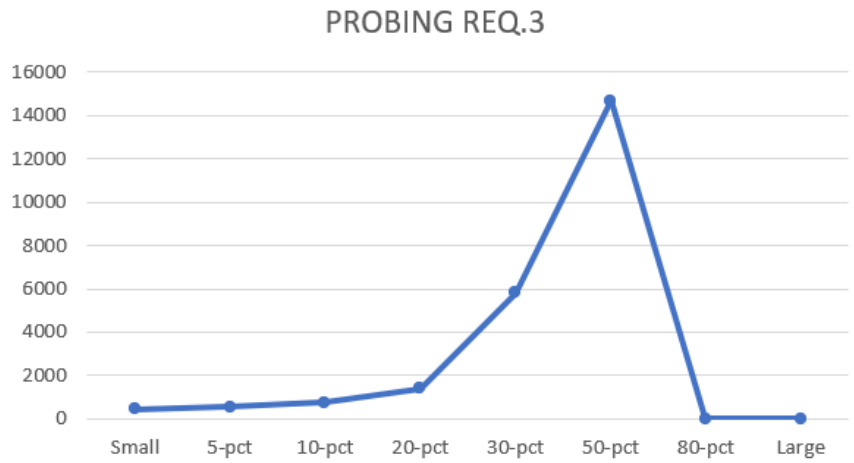
Chaining

Archivo	Tiempo (mseg)
Small	101.3
5-pct	573.2342
10-pct	834.437
20-pct	10467.75384
30-pct	1764.64738
50-pct	34748.234
80-pct	-
Large	-



Probing

Archivo	Tiempo (mseg)
Small	452.15
5-pct	534.75
10-pct	765.05
20-pct	1407.65
30-pct	5807.48
50-pct	14676.87
80-pct	-
Large	-



Como se observa en las gráficas y tablas correspondientes, usando probing se obtienen resultados mucho más eficientes en este requerimiento en cuestión del tiempo, pues incluso con los archivos de mayor tamaño posible (exceptuando 80pct y large) los tiempos son reducidos aproximadamente a la mitad.

Requerimiento 4. (Juan Manuel Pérez)

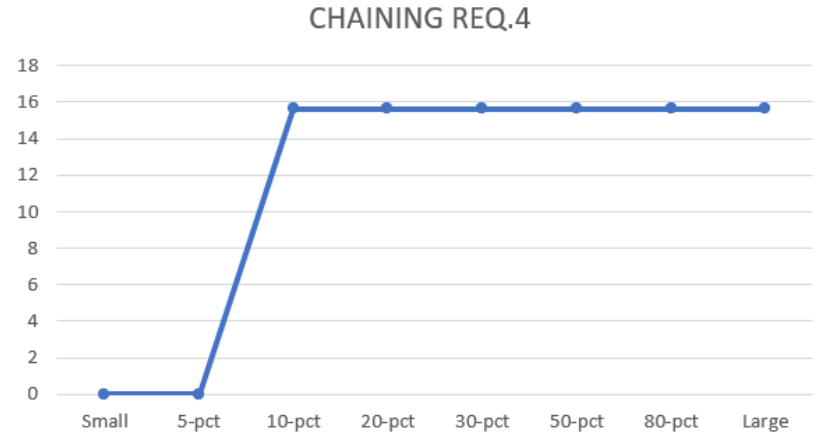
Tabla de Comparación de complejidad

	RETO 1	RETO 2
Complejidad	O(n log (n))	O(n log (n))

Tiempos de ejecución

CHAINING

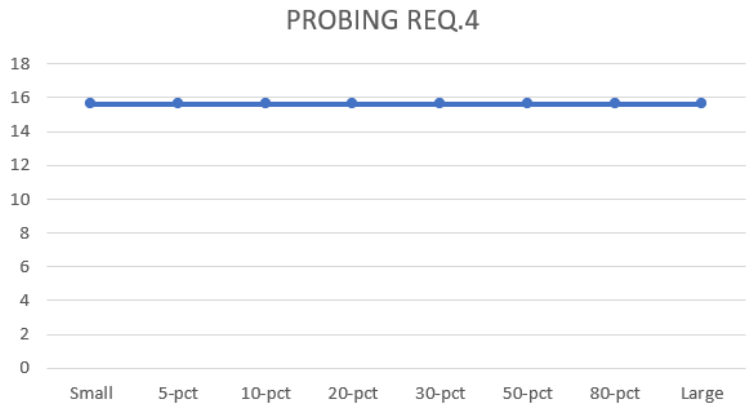
Archivo	**Tiempo (mseg)**
Small	0.0
5-pct	0.0
10-pct	15.625
20-pct	15.625



30-pct	15.625
50-pct	15.625
80-pct	15.625
Large	15.625

PROBING

Archivo	Tiempo (mseg)
Small	15.625
5-pct	15.625
10-pct	15.625
20-pct	15.625
30-pct	15.625
50-pct	15.625
80-pct	15.625
Large	15.625



Para el Requerimiento 4 la complejidad de igual forma se da en el Merge Sort. La complejidad de este algoritmo es de O(n log(n)). Dado los tiempos de ejecución los más rápidos es el de Chaining por lo que este fue el que se escogió.

Requerimiento 5.

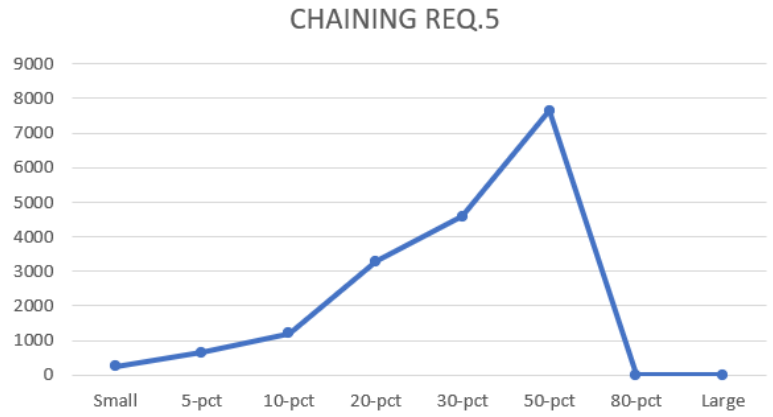
Tabla de Comparación de complejidad

	RETO 1	RETO 2
Complejidad		O(n log(n))

Chaining

Archivo	Tiempo (mseg)

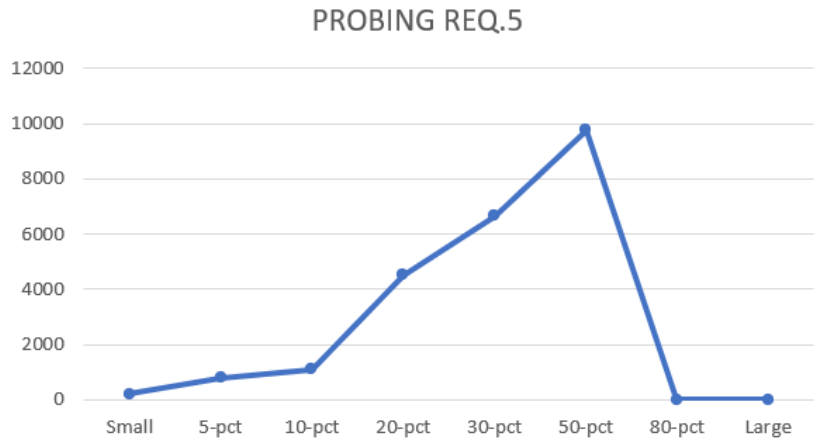
Small	245.10
5-pct	654.2
10-pct	1203.70
20-pct	3289.0
30-pct	4596.95
50-pct	7645.25
80-pct	-
Large	-



Probing

Archivo	Tiempo (mseg)
Small	202.45
5-pct	789.0
10-pct	1101.534
20-pct	4505.375
30-pct	6654.3
50-pct	9750.0
80-pct	-
Large	-

Como se puede observar en los datos tomados de los tiempos de ejecución, tanto usando probing



como chaining los resultados son similares. Sin embargo, resultan tiempos de ejecucion menores con el uso de map chaining, esto debido a la cantidad de datos y ciclos que se llevan a cabo en el requerimiento 5.