

Practica 0 Estructura de Datos y algoritmos 2

1. Tabla con los datos obtenidos después de usar los diferentes códigos para organizar un numero de palabras

Numero de datos	Tiempo Bubble Sort(Segundos)	Tiempo Bucket Sort(Segundos)
10000	0.4652489	0.0108416
20000	2.1471206	0.0132339
30000	4.2654803	0.021
40000	8.167294	0.0480382
50000	11.7506811	0.034141
60000	17.2151194	0.0356855
70000	24.3763492	0.0600672
80000	31.1422708	0.1040759
90000	39.9927833	0.0596249
100000	49.6414073	0.0719653
110000	60.1087232	0.0529014
120000	71.7731312	0.0665557
130000	84.0215275	0.0628736
140000	97.523025	0.0416013
150000	111.9962277	0.081597
160000	127.2922301	0.0939053
170000	143.661036	0.0975522
180000	170.854425	0.1372087
190000	208.995786	0.0929216
200000	245.114575	0.0916439
210000	278.174104	0.0857736
220000	300.192404	0.1085719
230000	331.169475	0.1644391
240000	383.7702427	0.1027691
247047	392.383643	0.1328677

2. Especificaciones de la máquina para hacer las pruebas

Procesador

AMD Ryzen 5 4500U with Radeon Graphics, 2375 Mhz, 6 Core(s), 6 Logical

Memoria

8.0 GB

Speed: 3200 MT/s

Disk (C:)

PM991 NVMe Samsung 256GB

Capacity: 239 GB

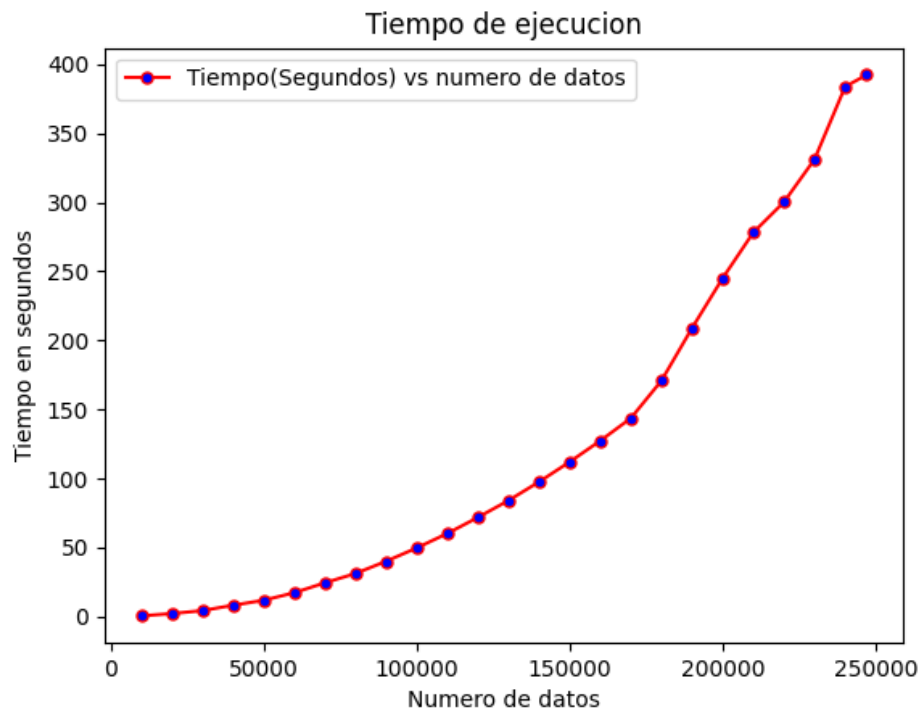
Formatted: 238 GB

Type: SSD

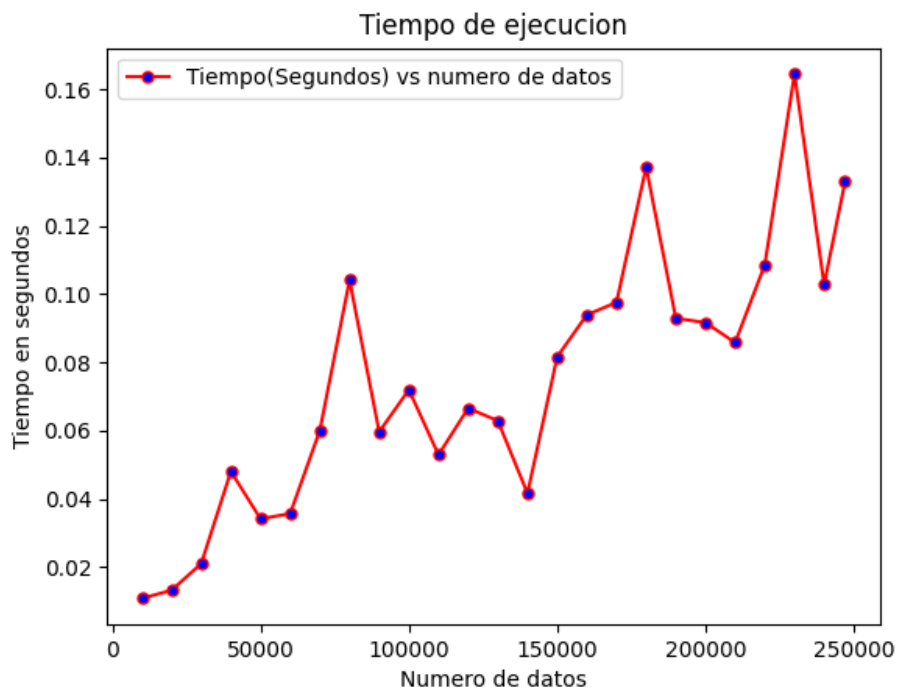
Read speed 49.4 KB/s

Write speed 45.3 KB/s

3. Información de las tablas anteriores en forma gráfica y análisis



Tablas de tiempo de ejecución del Bubble sort y Bucket sort



Análisis General

Comportamiento de Bubble Sort:

Tendencia al aumento: Es evidente que a medida que aumenta el número de datos, el tiempo que tarda Bubble Sort en ordenar la lista crece de manera significativa. Esto es característico de algoritmos de ordenamiento cuadráticos, como el Bubble Sort.

Crecimiento no lineal: El aumento en el tiempo no es proporcional al aumento en el número de datos, sino que crece a un ritmo mucho más acelerado.

Ineficiencia para grandes conjuntos de datos: Para grandes cantidades de datos, el Bubble Sort se vuelve extremadamente lento, lo que lo hace poco práctico para aplicaciones reales que requieran ordenar grandes conjuntos de datos.

Comportamiento de Bucket Sort:

Tendencia a un crecimiento moderado: A diferencia del Bubble Sort, el tiempo que tarda el Bucket Sort en ordenar la lista aumenta de manera mucho más lenta a medida que aumenta el número de datos.

Mayor eficiencia: Para la mayoría de los casos, el Bucket Sort es significativamente más rápido que el Bubble Sort, especialmente para grandes conjuntos de datos.

Fluctuaciones: Aunque la tendencia general es un crecimiento moderado, se observan algunas fluctuaciones en los tiempos, lo que podría deberse a la distribución de los datos dentro de los buckets.

Conclusiones

Bucket Sort es superior al Bubble Sort en este escenario: Los resultados obtenidos claramente muestran que Bucket Sort es un algoritmo de ordenamiento mucho más eficiente que el Bubble Sort para los datos proporcionados.

Complejidad algorítmica: Estos resultados corroboran las complejidades teóricas de ambos algoritmos. Bubble Sort tiene una complejidad promedio de $O(n^2)$, lo que explica su crecimiento cuadrático del tiempo de ejecución. Bucket Sort, en condiciones ideales, puede tener una complejidad promedio de $O(n+k)$, donde k es el número de buckets, lo que explica su crecimiento más lento.

Influencia de la distribución de los datos: La eficiencia de Bucket Sort depende en gran medida de la distribución de los datos. Si los datos están distribuidos uniformemente, Bucket Sort suele ser muy eficiente. Sin embargo, si los datos están agrupados en ciertos rangos, la eficiencia puede disminuir.