**Práctica 0**

**Tabla:**

|  |  |  |
| --- | --- | --- |
| Palabras | Bucket Sort | Bubble Sort |
| 10000 | 0,007 | 1,019 |
| 20000 | 0,01 | 3,874 |
| 30000 | 0,016 | 9,243 |
| 40000 | 0,022 | 16,540 |
| 50000 | 0,027 | 24,664 |
| 60000 | 0,053 | 37,401 |
| 70000 | 0,043 | 51,521 |
| 80000 | 0,047 | 62,641 |
| 90000 | 0,051 | 81,667 |
| 100000 | 0,067 | 102,363 |
| 110000 | 0,066 | 117,365 |
| 120000 | 0,06 | 142,474 |
| 130000 | 0,063 | 162,573 |
| 140000 | 0,091 | 190,953 |
| 150000 | 0,1 | 204,592 |
| 160000 | 0,109 | 218,231 |
| 170000 | 0,102 | 270,543 |
| 180000 | 0,116 | 286,457 |
| 190000 | 0,118 | 302,372 |
| 200000 | 0,13 | 350,321 |
| 210000 | 0,124 | 367,837 |
| 220000 | 0,135 | 391,876 |
| 230000 | 0,139 | 409,689 |
| 240000 | 0,145 | 557,448 |
| 247047 | 0,158 | 604,552 |

**Gráfica:**

**Especificaciones de la máquina utilizada:**

Tipo de procesador: AMD Ryzen 5 7535HS with Radeon Graphics

Velocidad: 3.30 GH

Capacidad memoria: 16 GB

Capacidad disco: 474 GB

**Análisis y conclusion:**

1. El bucket sort no solo es más eficiente, también permite manejar los datos de forma más personalizada.
2. Aunque el bubble sort tiene un funcionamiento más simple, esto limita algunas funciones que en el bucket sort se pueden realizar directamente.
3. La complejidad algorítmica del bubble sort es O(n^2)
4. La complejidad algorítmica del bucket sort es O(n^2) en el peor caso, pero este numero se reduce según el algoritmo de ordenamiento usado internamente en cada bucket (en este caso usé la función sort que usa heapsort, Quicksort e insertion sort) llegando a O(log n)
5. En este caso, el bucket sort es el algoritmo más eficiente por complejidad algorítmica.