# Comparación TrieSort y MergeSort

|  |  |  |
| --- | --- | --- |
| Datos | TrieSort | MergeSort |
| 1000 | 5710100 | 827900 |
| 2000 | 9624900 | 558300 |
| 3000 | 4046200 | 816200 |
| 4000 | 2658400 | 1149200 |
| 5000 | 3407400 | 1533900 |
| 6000 | 3347200 | 1500300 |
| 7000 | 5490000 | 1813800 |
| 8000 | 2635900 | 2139000 |
| 9000 | 3503100 | 2419800 |
| 10000 | 3316400 | 2788000 |
| 11000 | 3302500 | 3178900 |
| 12000 | 3052600 | 3076100 |
| 13000 | 3425700 | 3392200 |
| 14000 | 6430700 | 3683400 |
| 15000 | 6861500 | 4052400 |
| 16000 | 4102400 | 4491300 |
| 17000 | 4414800 | 4656000 |
| 18000 | 4653500 | 5109000 |
| 19000 | 5533600 | 5245400 |
| 20000 | 4790000 | 5490600 |
| 21000 | 6570800 | 5795100 |
| 22000 | 6277800 | 6162300 |
| 23000 | 7088700 | 6608200 |
| 24000 | 10155700 | 6827600 |
| 25000 | 10407000 | 7012200 |
| 26000 | 13144500 | 7477600 |
| 27000 | 7477100 | 7654400 |
| 28000 | 7743900 | 8984700 |
| 29000 | 8195800 | 9451200 |
| 30000 | 8671200 | 10390500 |
| 31000 | 8472800 | 10451500 |
| 32000 | 7417600 | 10491400 |
| 33000 | 9164300 | 11357600 |
| 34000 | 18337000 | 11459300 |
| 35000 | 9722300 | 12369100 |
| 36000 | 10468700 | 12185600 |
| 37000 | 14832400 | 12486800 |
| 38000 | 15088000 | 13003700 |
| 39000 | 15912200 | 13198700 |
| 40000 | 15439200 | 14250400 |
| 41000 | 8689200 | 14182800 |
| 42000 | 11401800 | 14496900 |
| 43000 | 11590500 | 13355100 |
| 44000 | 11914500 | 14143500 |
| 45000 | 12001900 | 14078100 |
| 46000 | 12050400 | 14432400 |
| 47000 | 12384800 | 14616100 |
| 48000 | 12645900 | 14979000 |
| 49000 | 12371100 | 15236600 |
| 50000 | 13328300 | 16876100 |

Al observar la grafica nos damos cuenta de que le trieSort tiene un comportamiento bastante errático, esto se debe a que la velocidad del trie depende de muchos factores además del número de datos, como la longitud de las palabras, que tantas comparten prefijos, entre otros factores. El mergeSort presenta resultados más lineales, si los comparamos con los resultados del trieSort podemos ver que por lo general es más rápido o esta muy cerca del trie, solo fue más lento los intervalos aproximados de 27000 a 36000 y de 40000 a 50000 y la diferencia no es demasiado grande considerando que los datos están dados en nanosegundos. En conclusión, el mergeSort nos presenta resultados más fáciles de predecir al mostrar un comportamiento lineal y es más rápido o igual que el trieSort en la mayoría de los casos. La manera en que están construidos los datos afectan mucho el rendimiento del trieSort, además de que la mayoría de las veces este algoritmo fue más lento, y con una gran diferencia, que el mergeSort.

El trieSort solo puede ser usado cuando los datos a ordenar son del tipo String, o si los elementos se comparan con una llave String, por lo que no puede ser usado con todo tipo de datos a diferencia de los demás algoritmos de ordenamiento estudiados en el curso los cuales solo necesitaban que los datos fueran comparables. En conclusión, el trieSort solo funciona para casos muy específicos y su rendimiento depende mucho de como estén construidos los datos.