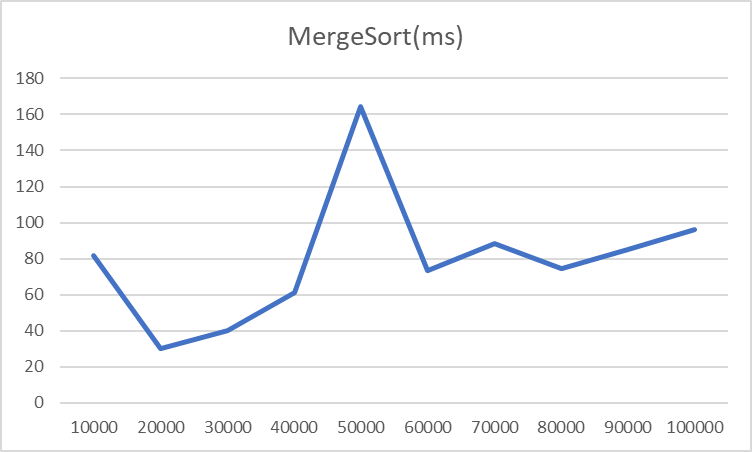
Estructuras de Datos Avanzadas

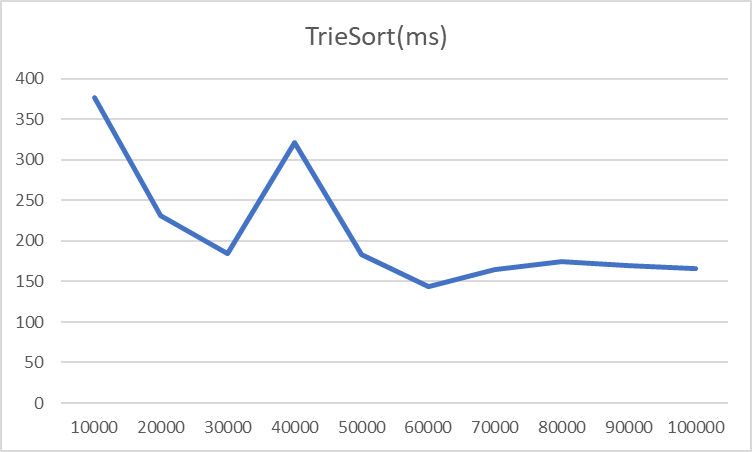
Tarea Tries – José Luis Gutiérrez Espinosa (179888)

Para este trabajo se implementó un algoritmo de ordenamiento de arreglos utilizando la estructura de datos MinHeap, la cual se elaboró con un arreglo de String. Para los experimentos se utilizó el mismo banco de palabras que para las pruebas del Trie Sort y se obtuvieron dichos resultados desde 10,000 hasta 100,000 palabras en periodos de 10,000 palabras. Los resultados son los siguientes:

Merge Sort:



Trie Sort:



Heap Sort:

Datos:

|  |  |  |  |
| --- | --- | --- | --- |
| Palabras | MergeSort(ms) | TrieSort(ms) | HeapSort(ms) |
| 10000 | 81.8266 | 376.7628 | 1220.7856 |
| 20000 | 29.9856 | 230.7049 | 4960.8882 |
| 30000 | 40.3737 | 184.8581 | 1028.11195 |
| 40000 | 61.0673 | 321.1301 | 18387.1421 |
| 50000 | 164.1382 | 183.6604 | 30812.6601 |
| 60000 | 73.4547 | 143.1389 | 105147.2551 |
| 70000 | 88.5841 | 165.1304 | 61780.4257 |
| 80000 | 74.2394 | 174.5003 | 8175.14199 |
| 90000 | 85.0156 | 169.8536 | 250020.72 |
| 100000 | 95.9362 | 166.4327 | 129369.8545 |

Analizando las gráficas y los datos resultantes, podemos concluir que el HeapSort no es un algoritmo eficiente comparado con los otros dos debido a su mucho mayor tiempo de ejecución además de su irregular comportamiento según el número de entradas.