

Algoritmos de Ordenamiento

Práctica 01

EDER ALONSO AMPUERO ATAMARI
HOWARD FERNANDO ARANZAMENDI MORALES
JOSE EDISON PEREZ MAMANI
HENRRY IVAN ARIAS MAMANI

Universidad Nacional San Agustín de Arequipa

Arequipa
August 20, 2022

Tabla de contenidos

1 Características

Equipo usado para pruebas

2 Algoritmos

Quick Sort

Merge Sort

Heap Sort

Tree Sort

Características del equipo

Las pruebas se realizaron en un solo equipo y con la misma data de prueba.

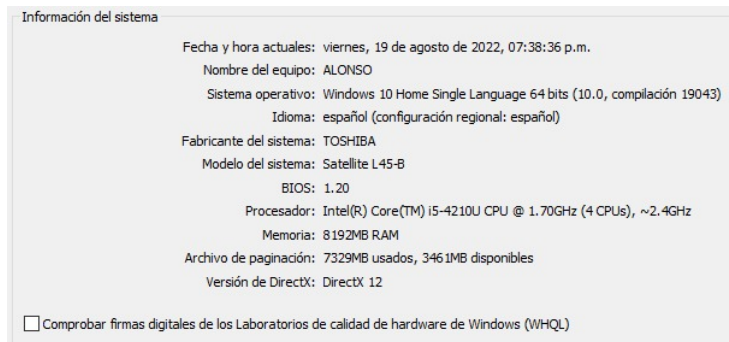


Figure: Características del equipo de pruebas

QUICK SORT

Quicksort ha sido históricamente el algoritmo genérico de ordenamiento más rápido conocido en la práctica. Es un algoritmo recursivo del tipo "divide y vencerás", y fácil de implementar.

Costo computacional: Caso promedio tarda $N \times \log N$, en el peor caso puede llegar a tardar N^2 .

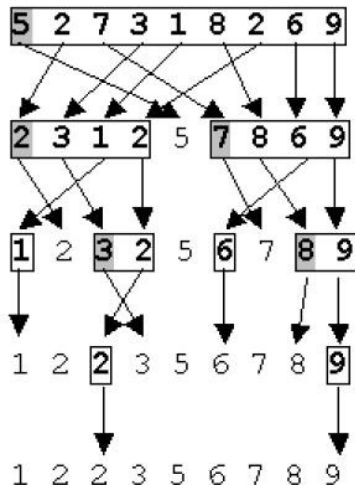


Figure: Quick Sort

Resultados

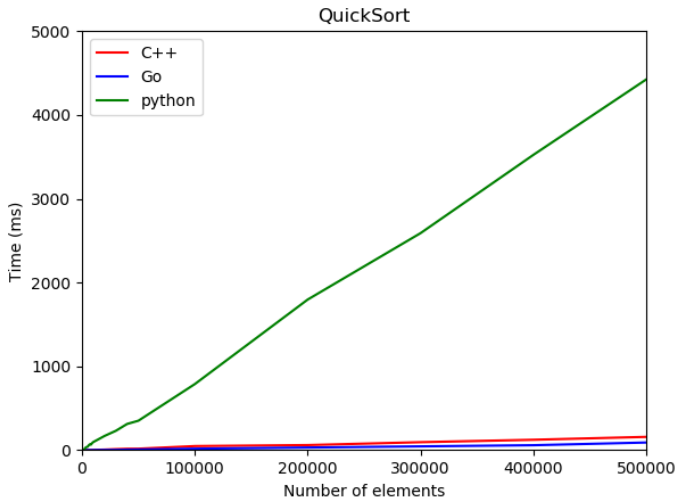


Figure: Quick Sort

MERGE SORT

- Es un algoritmo recursivo bastante eficiente para ordenar un array.
- Usa la técnica de divide y vencerás, la cual consiste en dividir el problema en sub problemas del mismo tipo que a su vez se dividirán hasta que sean suficientemente pequeños o triviales
- Costo computacional:
 $T(N) = N \log_2 N$

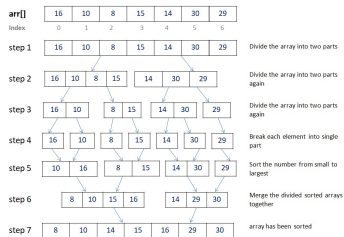


Figure: Merge Sort

Resultados

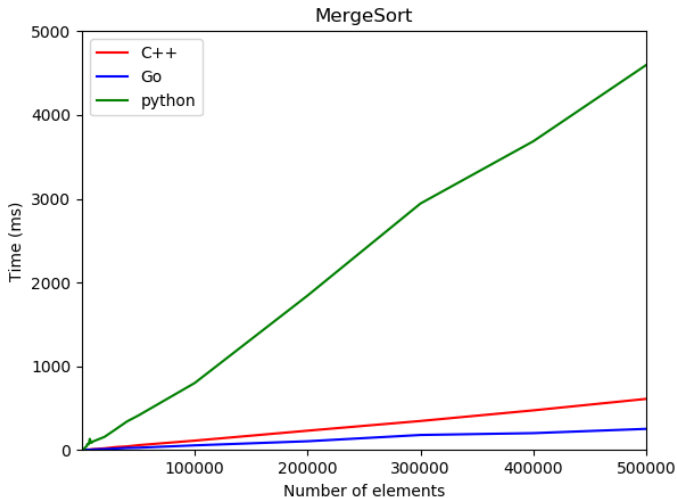


Figure: Merge Sort

HEAP SORT

Un heap es un objeto array que puede ser visto como un árbol binario.

El algoritmo de ordenamiento Heapsort ordena este array y tiene las siguientes características:

- $O(n \log n)$ incluso en el peor caso, como el merge sort.
- El ordenamiento es in-situ, como el insertion sort.
- Combina lo mejor de los dos algoritmos.

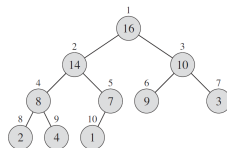


Figure: Heap Sort

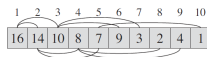


Figure: Heap Sort

Resultados

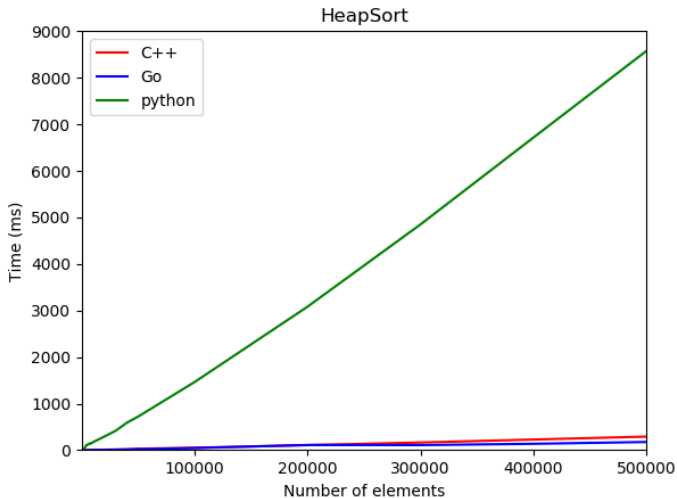


Figure: Heap Sort

TREE SORT

La clasificación de árbol es un algoritmo de clasificación que se basa en la estructura de datos del árbol de búsqueda binaria.

El algoritmo tiene las siguientes características

- $O(n \log n)$ para agregar y búsqueda
- El recorrido del árbol es inorden y como coste $O(n)$.

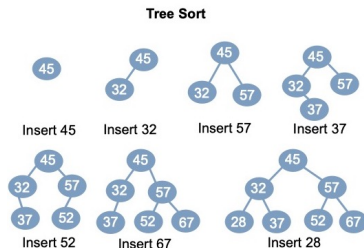


Figure: Tree Sort

Resultados

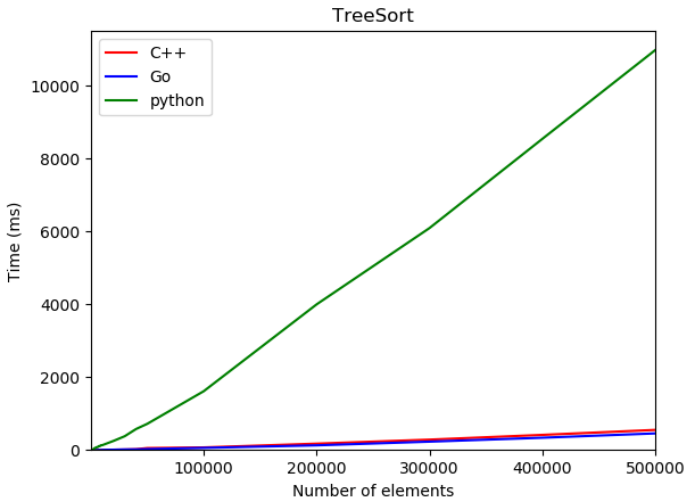


Figure: Tree Sort

Fin

Preguntas? Comentarios?