Universidad José Cecilio del Valle

Universidad José Cecilio Del Valle

**Proyecto II Parcial**

**Heap Sort Vs Merge Sort**

Alumno: Alessandro Galindo – 2017110421

Ingeniero: Kenneth Vittetoe

Clase: Estructura de datos

Fecha: 13 de noviembre del 2019

Índice

[Introducción 3](#_Toc24499173)

[Heap Sort u ordenamiento del montón 3](#_Toc24499174)

[Código 4](#_Toc24499175)

[Merge Sort u ordenamiento por mezcla 5](#_Toc24499176)

[Código 6](#_Toc24499177)

[Comparación de Algoritmos 7](#_Toc24499178)

[Se implementó un Arreglo de 10 para hacer la comparación 7](#_Toc24499179)

[Se implementó un arreglo de 50 para hacer la comparación 7](#_Toc24499180)

[Se implementó un arreglo de 100 para hacer la comparación 8](#_Toc24499181)

[Se implementó un arreglo de 900 para hacer la comparación 8](#_Toc24499182)

[Grafico Comparativo 9](#_Toc24499183)

[Conclusiones 10](#_Toc24499184)

[Bibliografía 11](#_Toc24499185)

# Introducción

Los algoritmos de ordenamiento se asemejan al trabajo que hace un registro, si bien sabemos, los registros generales están conformados de forma ordenada ya sea alfabéticamente o numéricamente dependiendo de que se trate este mismo.

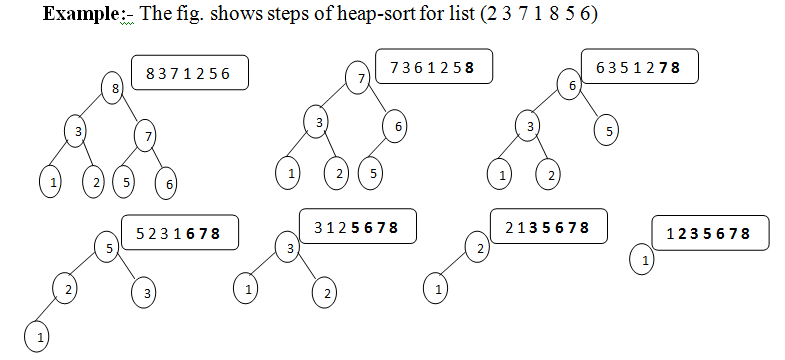
Estos Algoritmos nos ayudan o nos muestra un camino más fácil de hacer las cosas, estos son: El ordenamiento del monto (Heap Sort) y el ordenamiento de mezcla (Merge Sort).

En el siguiente informe se le estará detallando el funcionamiento de cada uno de estos métodos de ordenamiento con su respectiva definición y código en el lenguaje Java, así como también se le estará mostrando las diferentes pruebas realizadas con los códigos de cada uno y así hacer la comparación de estos dos métodos de ordenamiento.

# Heap Sort u ordenamiento del montón

La ordenación del montón es una técnica de clasificación basada en la comparación basada en la estructura de datos del montón binario. Es similar al orden de selección donde primero encontramos el elemento máximo y colocamos el elemento máximo al final. Repetimos el mismo proceso para el elemento restante.

¿Qué es el [montón binario](https://www.geeksforgeeks.org/binary-heap/) ?  
Primero definamos un árbol binario completo. Un árbol binario completo es un árbol binario en el que todos los niveles, excepto posiblemente el último, están completamente llenos, y todos los nodos están lo más a la izquierda posible (Fuente [Wikipedia](http://en.wikipedia.org/wiki/Binary_tree#Types_of_binary_trees) )

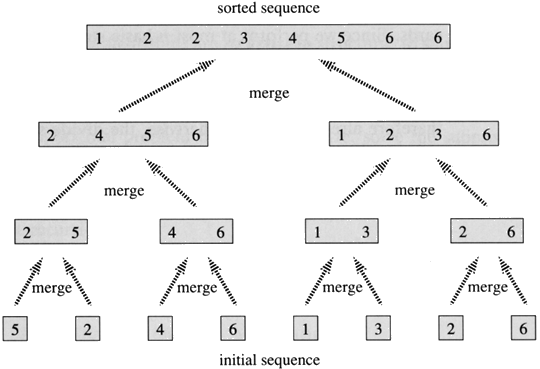
Un [montón binario](https://www.geeksforgeeks.org/binary-heap/) es un árbol binario completo donde los elementos se almacenan en un orden especial, de modo que el valor en un nodo principal es mayor (o menor) que los valores en sus dos nodos secundarios. El primero se llama montón máximo y el segundo se llama montón mínimo. El montón se puede representar por árbol binario o matriz.

## Código

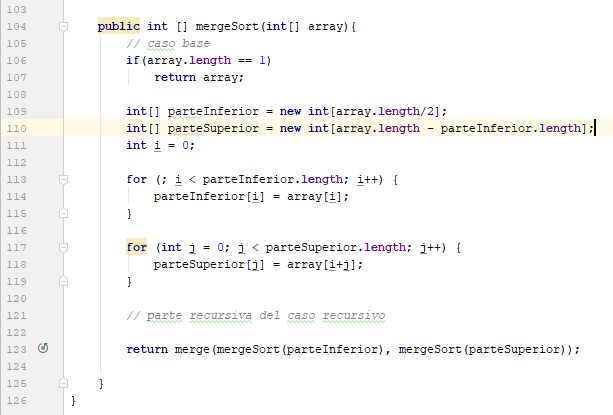
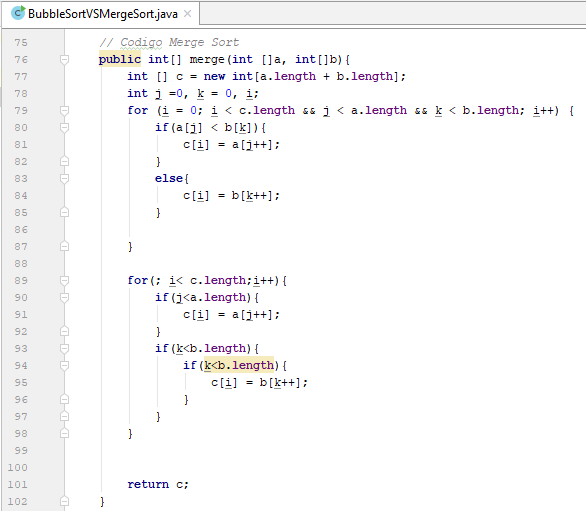
# Merge Sort u ordenamiento por mezcla

Es un algoritmo que utiliza la recursividad de un conjunto de elementos que se dividen entre dos. Se ordena cada parte de forma separada y al final se combinan los dos resultados en el arreglo original. También utiliza en principio “Divide y vencerás”.

Consiste en dividir el problema a resolver en subproblemas del mismo tipo que a su vez se dividirán, mientras no sean suficientemente pequeños o triviales.



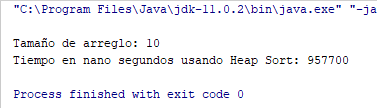
## Código



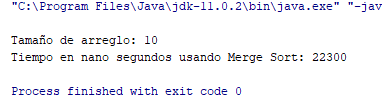
# Comparación de Algoritmos

## Se implementó un Arreglo de 10 para hacer la comparación

* **Heap Sort**

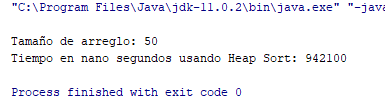


* **Merge Sort**

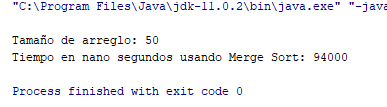


## Se implementó un arreglo de 50 para hacer la comparación

* **Heap Sort**

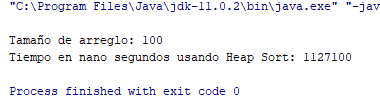


* **Merge Sort**

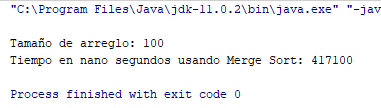


## Se implementó un arreglo de 100 para hacer la comparación

* **Heap Sort**

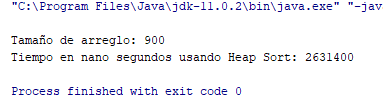


* **Merge Sort**

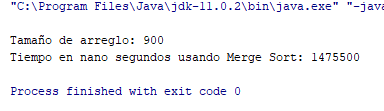


## Se implementó un arreglo de 900 para hacer la comparación

* **Heap Sort**



* **Merge Sort**



# Grafico Comparativo

# Conclusiones

Como conclusión a este proyecto podemos ver que a pesar de haber investigado un nuevo método de ordenamiento (Heap Sort) este no alcanzo a superar el tiempo que hizo el método de ordenamiento Merge Sort, este fue, de acuerdo a las pruebas realizadas el que supero por completo al Heap Sort entonces queda comprobado una vez más que la mejor opción es Merge Sort cuando queremos implementar un método de ordenamiento en nuestro programa.

Entonces podemos concluir y recomendar a los nuevos programadores que utilicen el ordenamiento de mezcla **Merge Sort** para hacer más eficiente su trabajo.

# Bibliografía

EcuRed (Merge Sort) - Recuperado el 14 de octubre de 2019, de

https://www.ecured.cu/MergeSort

GeeksForGeeks (Heap Sort) - Recuperado el 13 de noviembre del 2019, de

<https://www.geeksforgeeks.org/heap-sort/>

**GitHub** - https://github.com/Alessandro0919/ProyectoEstructura2.git