**Santiago Rodriguez Bernal – 202011182**

**Laboratorio 4**

* 1. **¿El comportamiento con relación al orden de crecimiento temporal de los algoritmos es acorde a lo enunciado teóricamente?**
  2. Sí, ya que lo que se pudo observar a través de estas pruebas es que efectivamente lo documentado en la literatura es cierto, pues las pruebas reflejaron que efectivamente cada algoritmo de ordenamiento se demoraba acorde a la teoría, en algunos casos no lo hacia pero se inferio que posiblemente se debía a que el ordenamiento en que se encontraban los datos posiblemente no fuera el mas beneficioso y provocara que fuera su peor caso, pero en general siempre se cumplió las pruebas con lo que enuncia la teoría.

• **¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas? Y • De existir diferencias, ¿a qué creen que se deben?**

* 1. Sí, demasiado notable los tiempos llegaban a ser muy altos entre una maquina y otra, esto principalmente por las diferentes características de cada uno y los posibles procesos que no conocemos que se ejecutan en el background, por lo tanto esto refuerza la importancia de documentar cada algoritmo de ordenamiento pues de otra manera seria muy difícil esclarecer cual es mejor, que otro pues el hecho de ejecutar código en distintas maquinas afecta demasiado el tiempo de ejecución.
  2. **• ¿Cuál Estructura de Datos (ARRAY\_LIST o SINGLE\_LINKED) funciona generalmente mejor si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?**

El ARRAY\_LIST es significativamente mejor que el SINGLE\_LINKED, pues no estamos agregando o removiendo datos de manera constante, que es algo que le cuesta bastante al arraylist, simplemente estamos organizándola información que hemos cargado, sin realizar alguna modificación en esta, además el ARRAY permite muchos mas fácil encontrar valores específicos sin tener que iterar por todo el archivo como sucede en el singlelinked.

* 1. **• Teniendo en cuenta las pruebas de tiempo de ejecución reportadas por los algoritmos de ordenamiento probados (iterativos y recursivos), proponga un listado de estos ordenarlos de menor a mayor teniendo en cuenta el tiempo de ejecución que toma ordenar las obras de arte.**

1. **1)**ShellSort
2. **2)**MergeSort
3. **3)**QuickSort
4. **4)I**nsertioSort
5. **5)**SelectionSort