# DANIEL SANTOS OBSERVACIONES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| % | Tamaño (Array\_list) | Insertion\_sort (ms) | Selection(ms) | Shell(ms) | Merge\_sort |
| 5 | 245 | 36.97 | 58.5 | 33.66 | 6.5 |
| 20 | 980 | 588.84 | 967.52 | 191.06 | 27.4 |
| 50 | 2451 | 3899.79 | 6167.97 | 560.81 | 79.35 |
| 100 | 4903 | 16203.72 | 25672 | 1436.84 | 174.92 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| % | Tamaño (Single\_linked) | Insertion\_sort (ms) | Selection(ms) | Shell(ms) | Merge\_sort |
| 5 | 245 | 634 | 591.62 | 90.35 | 20.26 |
| 20 | 980 | 41722.93 | 38218.62 | 2255.34 | 255.977 |
| 50 | 2451 | 687799 | NN | 15928 | 1530.21 |
| 100 | 4903 | NN | NN | 82718.1 | 6143.9 |

1. ¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?

Para la primera máquina se observa que los datos parecen lineales, sin embargo, el mejor ajuste es un ajuste exponencial, este resultado se debe a que los datos son muy pequeños y prevalece el comportamiento lineal, sin embargo, cuando son grandes volúmenes de datos, el comportamiento logaritmo comienza a prevalecer. Corresponde al mejor ajuste posible el ajuste exponencial y se observa que se comporta de acuerdo con la teoría.

1. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?

Sí, en las máquinas con procesadores más rápidos y con más RAM, el programa corre más rápido, sin embargo, se mantiene el comportamiento linearítmico.

1. De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

Se deben a si se están ejecutando tareas en segundo plano que puedan consumir recursos, y debido a lo anteriormente mencionado, a los procesadores y a la RAM.

Comparando se obtuvo la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| Algoritmo | Array\_list | Single\_linked |
| Merge | Mejor | Mejor |
| Quick | Peor | Peor |

a. ¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

Si solo se considera eso, lo mejor sería Merge sort, debido a que tiene menor tiempo de respuesta.

b. Para el escenario de ordenamiento de impuestos, teniendo en cuenta los resultados de tiempo reportados por todos los algoritmos de ordenamiento (iterativos y recursivos), proponga un ranking de los algoritmos de ordenamiento (de mayor eficiencia a menor - en relación con los tiempos de ejecución) para ordenar la mayor cantidad de impuestos.

|  |
| --- |
| Sort(menos tiempo a más) |
| Merge |
| Quick |
| Shell |
| Insertion |
| Selection |