Observaciones-lab5

Juan Manuel Rodriguez – 202013372

Juan Diego Cruz Vega – 202015077

Tabla 1

|  |  |  |
| --- | --- | --- |
|  | Maquina 1 | Maquina2 |
| Procesador | Intel(R) Core(TM) i5-9300H CPU @2.40GHz 2.40GHZ | Intel(R) Core(TM) i5-8265U CPU  @1,60GHz 1,80GHZ |
| Memoria (Ram) | 8,00 GB | 8,00 GB |
| Sistema operativo | Windows 10 Pro 64-bits | Windows 10 Pro 64-bits |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **T**amaño de la muestra (ARRAYLIST) | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 593,75 | 4125 | 125 | 31,25 | 46,88 |
| 2000 | 2765,625 | 4187,5 | 187,5 | 78,13 | 46,88 |
| 4000 | 9531,25 | 18546,875 | 430 | 156.25 | 140,63 |
| 8000 | 40625 | 78734,375 | 860 | 390,63 | 296,88 |
| 16000 | 200640,625 | 304843,75 | 1870 | 750,00 | 593,75 |
| 32000 |  |  | 3900 | 1359,38 | 1640,63 |
| 64000 |  |  | 10840 | 3125,00 | 3000,00 |
| 128000 |  |  | 23180 | 6390,63 | 5859,38 |
| 256000 |  |  | 64900 | 14593,75 | 12687,50 |
| 512000 |  |  | 100400 | 200,00 | 330,00 |

Tabla2

Los datos que no es registrados en la tabla no se pudieron calcular ya que el programa se demora demasiado tiempo en dar el resultado con ese método.

Tabla 3

|  |  |
| --- | --- |
| Algoritmo | ARRAT\_LIST |
| Quick Sort | 2707,50 |
| Merge Sort | 2464,25 |

Grafica 1

Grafica 2

Grafica 3

Preguntas:

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

El programa se comportó como se esperaba, teniendo en cuenta lo visto en clase. Cada al comparar el desempeño de cada algoritmo se pudo evidenciar que el mas eficaz fue el mergesort, sin embargo esto solo por unas milésimas de segundo comparado con el Quicksort. También se observó la poca eficiencia de selectionsort.

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

Si hubo una evidente diferencia en las maquinas en las cuales se probó el programa, la diferencia fue en cuanto a tiempos de hasta minutos de diferencia.

Las razones pueden ser el procesador, la cantidad de memoria RAM e incluso la cantidad de ventanas abiertas.

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

La estructura de datos que tuvo un mejor desempeño a la hora de correr el programa fue ArrayList

1. Para el caso analizado de ordenamiento de los videos, teniendo en cuenta los resultados de tiempo reportados por todos los algoritmos de ordenamiento estudiados (iterativos y recursivos), proponga un ranking de los algoritmos de ordenamiento (de mayor eficiencia a menor eficiencia en tiempo) para ordenar la mayor cantidad de videos.

Mejor

1: Shell Sort

2: Quick Sort

3: Merge Sort

4: Selection Sort

5: Insertion Sort

Peor