OBSERVACIONES DE LA PRACTICA

Martín Rincón Cod 201914114

Mariana Ruiz Cod 202011140

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40 GHz | Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.19 GHz |
| Memoria RAM (GB) | 16GB | 8GB |
| Sistema Operativo | Windows 10 64-bits | Windows 10 Pro 64 bits |

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

# **Maquina 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tamaño de la muestra (ARRAYLIST) | Insertion Sort [ms] | Selection Sort [ms] | Shell Sort [ms] | Quick Sort [ms] | Merge Sort [ms] |
| 1000 | 718 | 1046.88 | 31.25 | 31,25 | 31.25 |
| 2000 | 2640.63 | 3500 | 78.125 | 78,12 | 62.5 |
| 4000 | 10750 | 13218 | 171.88 | 265,63 | 125 |
| 8000 | 45609.38 | 57796.88 | 500 | 421,88 | 296.86 |
| 16000 | 192265.63 | 224562.5 | 1515.625 | 1140,62 | 750 |
| 32000 | 814968.75(>10min) | 919890.63 (>10min) | 2609.38 | 2328,13 | 1203.125 |
| 64000 |  |  | 6078.125 | 5500,00 | 2911.88 |
| 128000 |  |  | 15203.13 | 14359,38 | 9250 |
| 256000 |  |  | 32671.87 | 33750,00 | No responde |
| 375942 |  |  |  | 53968,75 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tamaño de la muestra (LINKED\_LIST) | Insertion Sort [ms] | Selection Sort [ms] | Shell Sort [ms] | Quick Sort [ms] | Merge Sort [ms] |
| 1000 | 60531.25 | 59437.5 | 2515.63 | 2453.13 | 1906,25 |
| 2000 | 519828.13 | 405203.13 | 12390.63 | 10515.63 | 10343,75 |
| 4000 | (>10min) | (>10min) | 75015.24 | 53203.12 | 42828,13 |
| 8000 |  |  | 238156 | 262765.60 | 192453,13 |
| 16000 |  |  | 1317234.38 (>10min) | >10min | 764140,63 |
| 32000 |  |  |  |  |  |
| 64000 |  |  |  |  |  |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | X |  |
| Quick sort | X |  |

Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

* Gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
  + Comparación de rendimiento ARRAYLIST.
  + Comparación de rendimiento LINKED\_LIST.
  + Comparación de rendimiento para Insertion Sort.
  + Comparación de rendimiento para Selection Sort.
  + Comparación de rendimiento para Shell Sort.
  + Comparación de rendimiento para MergeSort.
  + Comparación de rendimiento para QuickSort.

# **Maquina 2**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 709,38 | 869,79 | 46,88 | 31,25 | 36,46 |
| 2000 | 2809,38 | 3510,42 | 93,75 | 72,92 | 57,29 |
| 4000 | 11288,83 | 14057,29 | 203,13 | 140,63 | 135,42 |
| 8000 | 48583,33 | 57312,50 | 484,38 | 286,46 | 265,46 |
| 16000 | 189015,63 | 233781,25 | 1171,88 | 614,58 | 593,75 |
| 32000 | 773682,29 | 944520,83 | 2598,96 | 1380,21 | 1307,29 |
| 64000 |  |  | 6276,04 | 3442,71 | 2760,42 |
| 128000 |  |  | 15156,25 | 9911,46 | 5880,21 |
| 256000 |  |  | 35688,17 | 46286,46 | 12536,46 |
| 512000 |  |  |  |  |  |

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (LINKED\_LIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 46125,00 | 40619,79 | 2317,71 | 1593,75 | 223,96 |
| 2000 | 367057,29 | 329187,50 | 9859,38 | 8151,04 | 869,79 |
| 4000 | (>10 min) | (>10 min) | 47901,04 | 37020,83 | 3421,88 |
| 8000 |  |  | 220567,71 | 150682,29 | 13562,50 |
| 16000 |  |  | 1317234,38 | 659218,58 | 54192,71 |
| 32000 |  |  |  |  | 218890,63 |
| 64000 |  |  |  |  | 888437,50 |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | X |  |
| Quick sort | X |  |

Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

* Gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  + Comparación de rendimiento ARRAYLIST.
  + Comparación de rendimiento LINKED\_LIST.
  + Comparación de rendimiento para Insertion Sort.
  + Comparación de rendimiento para Selection Sort.
  + Comparación de rendimiento para Shell Sort.
  + Comparación de rendimiento para MergeSort.
  + Comparación de rendimiento para QuickSort.

# **Preguntas de análisis**

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

Sí, como podemos ver, en ambas tomas la regresion cuadratica se ajusta casi perfectamente al algoirtmo quicksort; esto es consistente con la teoria que nos dice que la complejidad de este algoritmo es O(n^2). Por otra parte, para el mergesort tenemos que el mejor ajuste que nos pudo dar nuestro software es tambien cuadratico; esto tienes sentido ya que para valores tan grandes de n, log(n) tiene derivada 1/n->0. La teoria nos dice que el mejor ajuste deberia ser linearitmico, si tomamos la derivada de esta funcion (1+log(n)), vemos que para valores grandes de n, esta tiende a un valor constante C (dlog(n)/dn=1/n->0) así que en realidad nuestra funcion sera casi lineal. Ambas regresiones de mergesort tienen un coeficiente que multiplica n^2 muy pequeño, esto representa el leve incremento logaritmico.

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

Sí, hay diferencias. En general, la máquina 2 realiza los ordenamientos en menor tiempo que la máquina 1.

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

Se deben principalmente a la diferencia en los recursos que pose cada maquina, y en los subprocesos que el sistema operativo ejecuta constantemente.

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

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.

Se entiende A>B si A es más eficiente en tiempo que B. No se pudo ejecutar todos los algoritmos para el máximo tamaño de N, así que se comparan para N=32000 ya que para ese punto ya se empiezan a notar las diferencia; es seguro asumir que la jerarquía se mantiene conforme se introducen más datos:

Mergesort>QuickSort>Shellsort>insertionSort>SelectionSort