OBSERVACIONES DE LA PRACTICA

Juan José Osorio (202017020)

Thais Tamaio Ramírez (202022213)

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | 3,1 GHz Intel Core i5 de dos núcleos | Intel(R) Core(TM) i7-3667U CPU @ 2.00GHz 2.50 GHz |
| Memoria RAM (GB) | 8 GB 2133 MHz LPDDR3 | 4.00 GB (3.90 GB usable) |
| Sistema Operativo | macOS Big Sur versión: 11.2.1 | Windows 10 Home 64-bits |

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

# **Maquina 1**

## **Resultados**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) |
| 1000 | 665,03 | 756,61 | 44,53 |
| 2000 | 2626,41 | 2985,29 | 91,40 |
| 4000 | 10759,13 | 12601,12 | 195,30 |
| 8000 | 44174,44 | 53291,52 | 482,21 |
| 16000 | 184837,87 | 213631,37 | 1057,72 |
| 32000 |  |  | 2460,17 |
| 64000 |  |  | 5808,17 |
| 128000 |  |  | 14018,36 |
| 256000 |  |  | 35920,55 |
| 512000 |  |  | 57490,68 |

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) |
| 1000 | 46673,74 | 41332,08 | 2314,42 |
| 2000 | 379918,78 | 341968,10 | 10910,84 |
| 4000 | 3113859,949 | 2726708.66 | 50537,60 |
| 8000 |  |  | 252813,92 |
| 16000 |  |  |  |
| 32000 |  |  |  |
| 64000 |  |  |  |
| 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) |
| Insertion sort | X |  |
| Selection sort | X |  |
| Shell sort | X |  |

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

## **Graficas**

* Cinco 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.

# **Maquina 2**

## **Resultados**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) |
| 1000 | 953,125 | 1125,0 | 62,5 |
| 2000 | 3859,375 | 4375,0 | 140,625 |
| 4000 | 15750,0 | 17625,0 | 312,5 |
| 8000 | 62859,375 | 73734,375 | 718,75 |
| 16000 | 272937,5 | 311406,25 | 1578,125 |
| 32000 |  |  | 3484,375 |
| 64000 |  |  | 8593,75 |
| 128000 |  |  | 20281,25 |
| 256000 |  |  | 53000,0 |
| 512000 |  |  | 84328,125 |

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) |
| 1000 | 62031,25 | 57390,625 | 3359,375 |
| 2000 | 542203,125 | 468906,25 | 14875,0 |
| 4000 | 4245234,38 | 3782546,875 | 67484,375 |
| 8000 |  |  | 364671,875 |
| 16000 |  |  |  |
| 32000 |  |  |  |
| 64000 |  |  |  |
| 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) |
| Insertion sort | X |  |
| Selection sort | X |  |
| Shell sort | X |  |

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

## **Graficas**

* Cinco 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.

# **Preguntas de análisis**

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

Con base en lo enunciado teóricamente con respecto al tiempo de ejecución y a los órdenes de crecimiento de los diferentes ordenamientos que se emplearon, tanto en el mejor de los casos como en el peor de los casos, el ordenamiento más rápido debería ser el de tipo Shell Sort. Por otro lado, con base en la teoría de las dos estructuras de datos que se utilizaron, se conoce que recorrer una lista para organizar los elementos de esta y cargar los datos con una ARRAY LIST es más rápido que con una SINGLE LINKED LIST. Esto se debe a que los elementos en una ARRAYLIST se encuentran en posiciones contiguas en memoria, por lo que acceder a cada uno de estos se puede conseguir con un tiempo constante. Por el contrario, al usar una SINGLE LINKED LIST los elementos no están en posiciones adyacentes, por lo que cada elemento debe conservar una referencia a la posición del elemento siguiente. Esto evita que recorrer la lista se pueda realizar en un tiempo constante y, al mismo tiempo, ocupa más memoria.

Al analizar el comportamiento de los algoritmos por medio de las gráficas obtenidas, se puede apreciar que tanto para ARRAY LIST como para SINGLE LINKED LIST la diferencia del tiempo de ejecución de cada ordenamiento puede llegar a ser muy significativa. Esto se debe a que las gráficas muestran con claridad que el ordenamiento Shell Sort y su respectivo orden de crecimiento conllevan a que el algoritmo se ejecute en un tiempo menor a el de los otros dos ordenamientos. Además, esta diferencia significativa se puede apreciar tanto en las gráficas de ARRAY LIST como en las de SINGLE LINKED LIST, ya que en ambas Shell ort presenta un comportamiento logarítmico, no como Insertion Sort y Selection Sort, que presentan un comportamiento potencial y polinomial, respectivamente. Por consiguiente, se demuestra que los tiempos de ejecución de Shell Sort siempre serán menores a los tiempos de Insertion Sort y Selection Sort, al ejecutar el algoritmo que se utilizó en las pruebas. Al mismo tiempo, al observar las gráficas y los valores obtenidos, se puede apreciar que para este algoritmo en específico la estructura de datos ARRAY LIST tuvo menores tiempos de ejecución que SINGLE LINKED LIST.

De igual manera, es gracias a los diferentes órdenes de crecimiento que tienen las dos estructuras de datos y los tres ordenamientos utilizados, que en las gráficas se puede apreciar con claridad que hay una diferencia significativa en la magnitud del tiempo de ejecución. Esto trae como consecuencia que una sola gráfica muestre comportamientos y valores contrastantes.

Teniendo en cuenta lo mencionado, el comportamiento de los algoritmos que pudimos observar a lo largo de las pruebas con distintos ordenamientos, distintas estructuras de datos y distintas cantidades de datos, es acorde a lo enunciado teóricamente. Es decir, que al realizar las pruebas se pudo apreciar que al utilizar la estructura de datos ARRAY LIST, el algoritmo cargaba y procesaba los datos mucho más rápido que cuando se utilizó SINGLE LINKED LIST. De igual manera, el algoritmo de ordenamiento más efectivo en todas las situaciones que fueron probadas fue Shell Sort.

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

diferentes máquinas?

Teniendo en cuenta los resultados al ejecutar el algoritmo con dos estructuras de datos diferentes (ARRAY LIST y SINGLE LINKED LIST), al utilizar tres tipos de ordenamientos (Insertion Sort, Selection Sort y Shell Sort) y al utilizar cantidades diferentes de datos, se pudo apreciar una diferencia significativa en el tiempo de ejecución del algoritmo para cada máquina.

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

Creemos que las diferencias en el tiempo de ejecución de los algoritmos se deben principalmente a la capacidad de RAM que tienen ambos computadores. Esto se debe a que uno de los computadores tiene 4 GB de RAM y el otro cuenta con 8 GB de RAM. Asimismo, el tiempo de ejecución que se aprecia en cada máquina puede depender también del procesador de cada computador e incluso el sistema operativo de estos. Por ejemplo, la capacidad de RAM de un computador puede asegurar que múltiples datos puedan ser cargados contiguamente en una ARRAY LIST de manera eficiente, al contar con suficiente espacio para guardarlos.

Por otro lado, consideramos que tanto la capacidad de RAM de los dos computadores utilizados en las pruebas, como el gran orden de crecimiento de ciertas estructuras de datos y ordenamientos utilizados, fue lo que generó tiempos muy altos para ciertas cantidades de datos. Con base en lo mencionado anteriormente, estos dos aspectos trajeron como consecuencia el no poder realizar todas las pruebas con los valores, ordenamientos y estructuras de datos propuestas. Por ende, se puede afirmar que tanto la capacidad de RAM de una máquina, como la eficiencia de una estructura de datos son aspectos muy importantes que pueden afectar significativamente el tiempo de ejecución de un algoritmo.

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

ejecución de los algoritmos?

Si solo se toman en cuenta los tiempos de ejecución de los algoritmos, se podría decir que el más eficiente en este aspecto, es la estructura de datos de tipo ARRAY LIST. Esto se debe, a que los tiempos de ejecución fueron menores (para este algoritmo en específico de encontrar videos con mayor número de vistas) con todos los ordenamientos utilizados y con diferentes cantidades de datos para procesar.