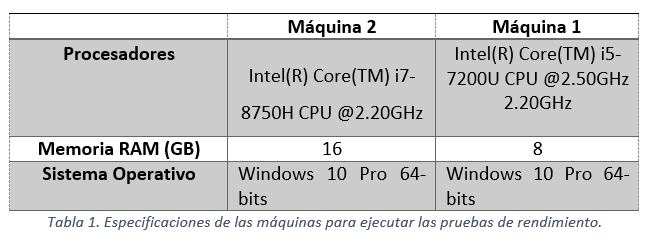
OBSERVACIONES DE LA PRACTICA 5

Natali Mercado Solórzano 202012682

Gabriel Villabon 202013898



Máquina 1

Nota: Las tablas se llenaron hasta donde la máquina respondió en menos de 12 minutos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tamaño de la muestra (array\_list) | Insertion Sort [ms] | Selection Sort [ms] | Shell Sort [ms] | Quick Sort [ms] | Merge Sort [ms] |
| 1000 | 640,625 | 828,125 | 62,5 | 46,875 | 62,5 |
| 2000 | 3140,625 | 3375 | 109,375 | 46,875 | 78,125 |
| 4000 | 12109,375 | 14671,875 | 281,25 | 203,125 | 171,875 |
| 8000 | 72375 | 60828,125 | 515,625 | 328,125 | 328,125 |
| 16000 |  | 228140,625 | 1296,875 | 812,5 | 593,75 |
| 32000 |  |  | 3046,875 | 1765,625 | 1390,625 |
| 64000 |  |  | 8031,25 | 3875 | 3203,125 |
| 128000 |  |  | 21453,125 | 7906,25 | 11937,5 |
| 256000 |  |  |  | 15843,75 | 15687,5 |
| 375942 |  |  |  | 23296,875 | 23296,875 |
| 512000 |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tamaño de la muestra (linked\_list) | Insertion Sort [ms] | Selection Sort [ms] | Shell Sort [ms] | Quick Sort [ms] | Merge Sort [ms] |
| 1000 |  | 71625 | 3359,375 | 359,375 | 2906,25 |
| 2000 |  |  | 19390,625 | 1406,25 | 12359,375 |
| 4000 |  |  | 87125 | 7890,625 | 66484,375 |
| 8000 |  |  |  | 22984,375 | 395000 |
| 16000 |  |  |  | 109406,25 |  |
| 32000 |  |  |  |  |  |
| 64000 |  |  |  |  |  |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

Máquina 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tamaño de la muestra (array\_list) | Insertion Sort [ms] | Selection Sort [ms] | Shell Sort [ms] | Quick Sort [ms] | Merge Sort [ms] |
| 1000 | 453,125 | 468,75 | 46,875 | 15,625 | 15,625 |
| 2000 | 1968,75 | 1953,125 | 62,5 | 62,5 | 46,875 |
| 4000 | 7796,875 | 8015,625 | 140,625 | 93,75 | 109,375 |
| 8000 | 35796,875 | 34390,625 | 328,125 | 187,5 | 203,125 |
| 16000 | 159312,5 | 152171,875 | 765,625 | 500 | 406,25 |
| 32000 | 564984,375 | 647359,375 | 1953,125 | 1078,125 | 843,75 |
| 64000 | 2595203,13 | 2509546,88 | 4625 | 2593,75 | 1968,75 |
| 128000 |  |  | 12218,75 | 4546,875 | 4328,125 |
| 256000 |  |  | 28031,25 | 9843,75 | 9578,125 |
| 375942 |  |  |  | 13171,875 | 15234,375 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tamaño de la muestra (linked\_list) | Insertion Sort [ms] | Selection Sort [ms] | Shell Sort [ms] | Quick Sort [ms] | Merge Sort [ms] |
| 1000 | 40171,875 | 35812,5 | 2484,75 | 312,5 | 1468,75 |
| 2000 | 327687,5 | 295734,375 | 9531,25 | 718,75 | 8000 |
| 4000 | 2484,75 | 2460906,25 | 46562,5 | 3750 | 38656,25 |
| 8000 |  |  | 22465,625 | 12812,5 | 138093,75 |
| 16000 |  |  | 1020578,13 | 56312,5 |  |
| 32000 |  |  |  | 265109,375 |  |
| 64000 |  |  |  |  |  |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

Actividad

1. Diagrame tres (2) gráficas comparativas para cada uno de los algoritmos de ordenamiento.
2. Para las gráficas diagrame el eje X (1000 a 512000 elementos) según el tamaño de la muestra y el eje Y (ms) según el tiempo de ejecución de los algoritmos.

3) Calcule las líneas de tendencia más apropiadas para los datos en cada una de las gráficas.

4) Compare los resultados obtenidos con la complejidad teórica de cada algoritmo y con los resultados obtenidos de las pruebas en ambas maquinas respondiendo las siguientes preguntas:

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

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

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

Desarrollo

1. 2) 3)

MergeSort en array\_list

QuickSort en array\_list

MergeSort en linked\_list

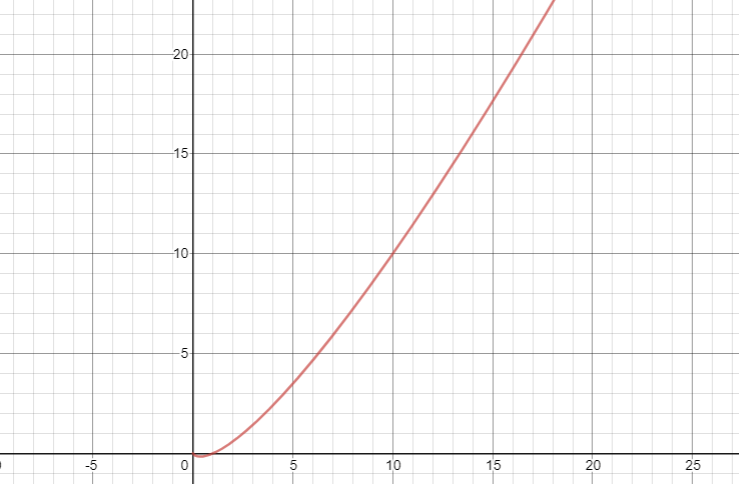
QuickSort en linked\_list

1. Teniendo en cuenta que la complejidad teórica para QuickSort en el mejor de los casos es O (nlog(n)) y en el peor de los casos es O(), se puede decir que en la mayoría de las pruebas de la práctica el tipo de ordenamiento se enfrentó al peor de los casos puesto que las gráficas tienden a una función potencial y es acorde a la teoría.

Por otro lado, la complejidad teórica de MergeSort en el mejor y peor de los casos es O (nlog(n)). Además, es un tipo de ordenamiento más útil cuando se trata de linked lists y es un tipo de ordenamiento estable, es decir, los elementos de un array o una matriz mantienen sus posiciones. Por lo anterior, se puede decir que los resultados no concuerdan del todo con la teoría pues en array se obtuvo una tendencia lineal y no linearitmica, sin embargo, esto puede deberse a que las posiciones se mantienen constantes. Por otro lado, en linked list la gráfica se asemeja a la esperada, por ende, concuerda con la teoría.

Gráfica teórica:

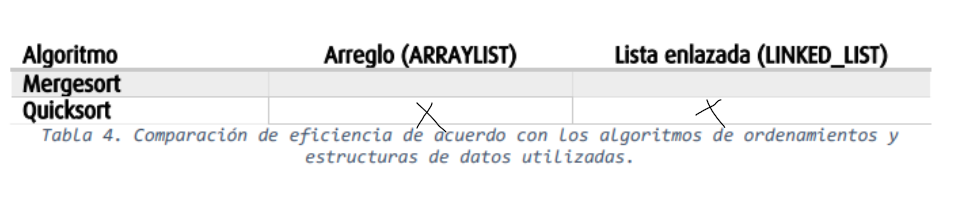
N (log(N))



1. Cree dos (2) gráficas comparativas para los rendimientos de los algoritmos de ordenamiento por cada tipo de representación. Una para arreglo (ARRAY\_LIST) y otra para la lista enlazada (LINKED\_LIST).
2. Compare los resultados utilizando estas dos graficas comparativas.

QuickSort y MergeSort tienen casi la misma complejidad cuando se trata de array\_list, sin embargo, en linked\_list QuickSort es casi constant y por ende más rápido en tiempo de ejecución.

1. Tomando como base las gráficas complete la Tabla 4 en el documento Observacioneslab5.docx e indique cual algoritmo es más eficiente en cual representación.



A partir de la Tabla 4 responda la siguiente pregunta en el documento Observaciones-lab5.docx: • ¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

QuickSort teniendo en cuenta la tabla y la gráfica.

9) Realice la comparación de los resultados obtenidos con los algoritmos iterativos analizados en el laboratorio anterior y respondan la siguiente pregunta: • 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.

1. Quick

2. Merge

3. Shell

4. Selection

5. Insertion