

Observaciones Laboratorio 4

Nombres con códigos:

Juan Felipe Serrano 201921654

Santiago Sinisterra 202022177

Tabla 1: Especificaciones de las máquinas

	Máquina 1	Máquina 2
Procesadores	Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.60 GHz	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz
Memoria RAM (GB)	16.0 GB	8.00 GB
Sistema Operativo	Windows 10 Home 64-bits	Windows 10 Home 64-bits

Tabla 2: Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo Estudiante 1

Porcentaje de la muestra	Tamaño de la muestra (ArrayList)	Insertion Sort(ms)	Shell Sort(ms)	Merge Sort (ms)	Quick sort (ms)
Small	768	234.375	15.625	15.625	15.625
05.00%	7572	22687.5	250.0	156.25	406.25
10.00%:	15008	94406.25	578.125	406.25	1312.6
20.00%	29489	388958.7	1171.875	906.25	5187.5
30.00%	43704	N/A	1828.125	1062.5	N/A
50.00%	71432	N/A	2734.375	2218.75	N/A
80.00%	11781	N/A	4921.875	3828.125	N/A
Large	138150	N/A	7546.875	3937.5	N/A

Tabla 3: Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo Estudiante 2

Porcentaje de la muestra	Tamaño de la muestra (ArrayList)	Insertion Sort(ms)	Shell Sort (ms)	Merge Sort (ms)	Quick Sort (ms)
Small	768	406.25	31.25	31.25	31.25
05.00%	7572	38781.25	328.13	203.13	312.50
10.00%	15008	157687.50	796.88	448.38	640.63
20.00%	29489	649093.75	1515.63	12625.63	1796.88
30.00%	43704	NA	2956.25	1906.25	2281.25
50.00%	71432	NA	4171.88	3125.0	4828.13
80.00%	111781	NA	NA	4453.13	6515.63
Large	138150	NA	NA	5812.5	11718.75

Tabla 4: Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada Estudiante 1.

Porcentaje de la muestra	Tamaño de la muestra (Linked_list)	Insertion Sort(ms)	Shell Sort (ms)	Merge Sort (ms)	Quick Sort (ms)
Small	768	15484.375	843.75	171.875	1468.75
0.5.00%	7572	N/A	134578.125	239562	12609.375
10.00%	15008	NA	NA	NA	NA
20.00%	29489	NA	NA	NA	NA
30.00%	43704	NA	NA	NA	NA
50.00%	71432	NA	NA	NA	NA
80.00%	111781	NA	NA	NA	NA
Large	138150	NA	NA	NA	NA

Tabla 5: Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada Estudiante 2.

Porcentaje de la muestra	Tamaño de la muestra (Linked_list)	Insertion Sort(ms)	Shell Sort (ms)	Merge Sort (ms)	Quick Sort (ms)
Small	768	23390.63	937.5	140.63	1046.86
0.5.00%	7572	NA	187875.0	15843.75	NA
10.00%	15008	NA	NA	62890.63	NA
20.00%	29489	NA	NA	NA	NA
30.00%	43704	NA	NA	NA	NA
50.00%	71432	NA	NA	NA	NA
80.00%	111781	NA	NA	NA	NA
Large	138150	NA	NA	NA	NA

Preguntas de análisis:

1. ¿El comportamiento con relación al orden de crecimiento temporal de los algoritmos es acorde a lo enunciado teóricamente? Si los algoritmos se comportan muy parecido a sus valores teóricos de $O(n)$, sin embargo comparando shell y merge pudimos ver que en este caso para estos datos merge tiene un mejor desempeño.
2. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas? Si gracias a que la máquina del estudiante dos tiene un procesador menos potente y la mitad de ram creemos que es la razón por la que este se le demoran más en general todas los ordenamientos. Esto debido a que un mejor computador puede efectuar más operaciones por segundo haciendo que su desempeño sea mejor, y así reduciendo tiempo de espera en los ordenamientos. Interesantemente también encontramos que quick sort se comportaba muy distinto en ambos computadores ya que en el computador del estudiante 2 este era el segundo mejor, y en el computador del estudiante 1 este era el segundo peor y las diferencias son bastante significativas.

3. De existir diferencias, ¿a qué creen que se deben?

Como lo explicamos anteriormente creemos que estas diferencias se deben a la cpu y a la capacidad de memoria ram ya que el sistema operativo era el mismo y vimos que en el estudiante 2 los datos crecen más rápidamente. Más allá, al investigar la naturaleza de nuestras Cpus nos dimos cuenta que la del estudiante dos es de solo 2 cores mientras que la del estudiante 1 es de 4, esto también podría explicar la disparidad de resultado, además de la edad del computador del estudiante 2 siendo mucha más tiempo. No solo eso pero al revisar el administrador de tareas nos dimos cuenta que python y visual code Studio no consumen todo el procesamiento de cpu o la memoria ram, nunca pasa a más de 20% de consumo en la cpu y 50% en la memoria del estudiante 1, en otras palabras deberían estar limitadas las capacidades para el programa y esto también es independiente para cada computador por lo que es otra variable que podría afectar los resultados.

Tabla 6: Comparación de eficiencia de acuerdo con los algoritmos de ordenamiento y estructuras de datos utilizadas.

Algoritmo	Arreglo(ArrayList)	Lista Enlazada(Linked List)
Insertion Sort	4	4
Shell Sort	2	2
Merge Sort	1	1
Quick Sort	3	3

4. ¿Cuál Estructura de Datos (ARRAY_LIST o SINGLE_LINKED) funciona generalmente mejor si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

Funciona mejor Array_List gracias a que sus tiempos de ejecución para todos los algoritmos tiende a ser menor, también es importante mencionar como en linked list estas se dejan de tener un tiempo de ejecución decente desde 30% lo que nos deja ver que ni siquiera con merge(El cual es el organizador más eficiente en este ejemplo) no logra ordenar los datos en su totalidad en un tiempo decente.

5. Teniendo en cuenta las pruebas de tiempo de ejecución reportadas por los algoritmos de ordenamiento probados (iterativos y recursivos), proponga un listado de estos ordenarlos de menor a mayor teniendo en cuenta el tiempo de ejecución que toma ordenar las obras de arte.

Según los resultados del laboratorio se podría decir que el orden de menor a mayor en consumo de tiempo sería: