1. OBSERVACIONES DE LA PRACTICA

Daniel Felipe Vargas Cod 202123899

Estudiante 2 Cod XXXX

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel Celeron N3160 | 11th Gen Intel(R) Core(TM) i5-1135G7 |
| Memoria RAM (GB) | 1,889 GB | 16,0 GB |
| Sistema Operativo | Arch Linux x86\_64 | Microsoft Windows 10 Home Single Language |

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

# **Maquina 1**

## **Resultados**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (ARRAY\_LIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** |
| 0.50% | 501 | 581.60 | 628.97 | 52.14 |
| 5.00% | 4391 | 45929.23 | 50200.80 | 759.91 |
| 10.00% | 8193 | 160861.00 | 174861.06 | 1581.12 |
| 20.00% | 15136 | +600000 | +600000 | 3355.22 |
| 30.00% | 21319 | +600000 | +600000 | 4951.66 |
| 50.00% | 32422 | +600000 | +600000 | 8559.63 |
| 80.00% | 47165 | +600000 | +600000 | 12664.83 |
| 100.00% | 56129 | +600000 | +600000 | 15466.12 |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (ARRAY\_LIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** |
| 0.50% | 501 | 19624.32 | 17846.38 | 1471.03 |
| 5.00% | 4391 | +600000 | +600000 | 271055.94 |
| 10.00% | 8193 | +600000 | +600000 | +600000 |
| 20.00% | 15136 | +600000 | +600000 | +600000 |
| 30.00% | 21319 | +600000 | +600000 | +600000 |
| 50.00% | 32422 | +600000 | +600000 | +600000 |
| 80.00% | 47165 | +600000 | +600000 | +600000 |
| 100.00% | 56129 | +600000 | +600000 | +600000 |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAY\_LIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* |  |  |
| *Selection Sort* |  |  |
| *Shell Sort* | X | X |

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

# **Maquina 2**

## **Resultados**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (ARRAY\_LIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** |
| 0.50% | 501 | 845.96 | 847.82 | 68.40 |
| 5.00% | 4391 | 65180.91 | 65741.79 | 1021.54 |
| 10.00% | 8193 | 231126.72 | 233613.80 | 2148.11 |
| 20.00% | 15136 | 807943.07 | 856497.83 | 4489.16 |
| 30.00% | 21319 | - | - | 6852.61 |
| 50.00% | 32422 | - | - | 11737.12 |
| 80.00% | 47165 | - | - | 16859.20 |
| 100.00% | 56129 | - | - | 20993.96 |

Tabla 5. Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (SINGLE\_LINKED)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** |
| 0.50% | 501 | 10803.28 | 9719.19 | 324.84 |
| 5.00% | 4391 | +600000 | +600000 | 113302.38 |
| 10.00% | 8193 | +600000 | +600000 | 454785.68 |
| 20.00% | 15136 | +600000 | +600000 | +600000 |
| 30.00% | 21319 | +600000 | +600000 | +600000 |
| 50.00% | 32422 | +600000 | +600000 | +600000 |
| 80.00% | 47165 | +600000 | +600000 | +600000 |
| 100.00% | 56129 | +600000 | +600000 | +600000 |

Tabla 6. Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAY\_LIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* | O(2E+07 N^2) | O(1E+08N^1.7446) |
| *Selection Sort* | O(2E+07 N^2) | O(1E+08N^1.7905) |
| *Shell Sort* | O(23756 N^1.0806)  X – más eficiente | O(5E+07 N^2)  X – más eficiente |

Tabla 7. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

# **Preguntas de análisis**

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

Si. Pudimos encontrar esto al hacer una regresión sobre las curvas generadas por el tiempo que tardó cada ordenamiento en terminar, con lo cual pudimos ver que una función cuadrática se ajusta mejor a la gráfica de complejidad del insertion y selection sort (aunque no tuvimos la oportunidad de verificarlo también para la estructura SINGLE\_LINKED, ya que solo obtuvimos un dato con estos ordenamientos). EN el caso del ordenamiento de tipo Shell, encontramos que probablemente no encontramos suficientes datos para mostrar la naturaleza logarítmica que posee la complejidad de este ordenamiento, lo cual resultó en Excel frecuentemente prefiriendo una regresión lineal para ajustarse mejor a la gráfica de complejidad del Shell sort. Sin embargo, a pesar de esta situación, todavía es evidente como la complejidad de este ordenamiento no es lineal, y al realizar un ajuste del tipo x log(x) (para copiar la complejidad teórica del Shell sort de n log (n)), encontramos que esta también se ajusta casi perfectamente a la gráfica.

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

Si bien ejecutar las pruebas en una mejor maquina con más capacidad de procesamiento nos permite obtener más datos y por lo tanto realizar mejores aproximaciones, la complejidad de los algoritmos se mantiene en diferentes maquinas como es de esperar, independiente de su capacidad de procesamiento. Es aparente que ejecutar las pruebas en máquinas diferentes resultará en tiempos diferentes, pero lo importante es la relación entre los tiempos y el tamaño de muestra que nos permite determinar la complejidad de un algoritmo independientemente de la maquina donde se probó.

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

Como se señaló previamente, dichas diferencias se deben a las diferencias en capacidad de procesamiento que tienen diferentes maquinas. Principalmente, encontramos que lo que más impacta a la duración de un ordenamiento es la velocidad del reloj de la CPU con la cual se está ejecutando el algoritmo. Esto tiene sentido, ya que un a CPU que pueda realizas más ciclos por segundo podrá simplemente realizar más tareas en una misma cantidad de tiempo, en este caso realizar más comparaciones cada segundo, lo cual lógicamente reducirá el tiempo que tarda en ordenarse la totalidad de los datos.

Adicionalmente, la cantidad de memoria RAM disponible en cada computador tuvo un efecto visible, ya que en computadores con memora RAM más limitada en ocasiones era imposible completar ciertos ordenamientos, lo cual no era un problema en máquinas con más memoria RAM disponible que podían completar el ordenamiento sin quedarse sin memoria.

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

Si sólo se tiene en cuenta el tiempo de ejecución, es evidente que, entre los tres algoritmos analizados, shell es el mejor para utilizar. Esto debido a que tomó un mejor tiempo de ejecución que los otros dos en todas las pruebas que se pudieron realizar. Incluso, fue posible realizar la prueba de 20% en SINGLE\_LINKED para Shell en menos de 10 minutos, algo que los otros dos algoritmos no pudieron conseguir con la prueba de 5%. En ARRAY\_LIST la diferencia fue incluso más notoria, pues shell pudo completar todas las pruebas satisfactoriamente, con un tiempo máximo de 20993.96 ms para la prueba del 100% de los datos. Por otro lado, los otros dos algoritmos no pudieron ser evaluados más allá de la prueba del 20% (prueba en la que, por cierto, demoraron ya un poco más de 10 minutos). Cabe resaltar que desde la prueba del 5% insertion y selection ya tomaban un tiempo muy superior al tiempo máximo de shell. En definitiva, es claro que, si sólo se toma el tiempo de ejecución como medida de eficiencia, shell es el algoritmo más apropiado para usar.