Análisis del Reto 1

Estudiante 1 Sergio Rincón Cod 201914107

Estudiante 2 Luis Tejón Cod 202113150

El estudiante 1 desarrolló el requerimiento 3.

El estudiante 2 desarrolló el requerimiento 4.

* Carga de Datos:
* Requerimiento 1:
* Requerimiento 2:

**Tipo de TAD usado:** TAD lista (array\_list)

**Carga de datos:** Para este requerimiento se agregó en la carga inicial de datos un ordenamiento de tipo Merge para crear una sub lista de las obras, pero ordenada por la fecha de adquisición.

**Complejidad agregada a la carga de datos:** con n el número de obras.

**Requerimiento:** En el requerimiento al ya estar organizada la lista, hizo falta únicamente el uso de 2 búsquedas binarias para hallar las posiciones de la menor obra con la fecha inicial (o la menor obra con una fecha inmediatamente mayor) y la posición de la mayor obra con la fecha final (o la inmediatamente menor) con lo que, al ser de tipo array se pudo “cortar” la lista a partir de esas posiciones en O (1). Al final en la lista resultante se hizo una iteración para ver cuántas obras son adquiridas por compra

* Requerimiento 3:
* Requerimiento 4:
* Requerimiento 5:
* Requerimiento 6:

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz | Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 2.61GHz |
| Memoria RAM (GB) | 16GB | 12GB |
| Sistema Operativo | Windows 10 home 64-bits | Windows 10 home 64-bits |

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

# **Maquina 1 Resultados**

|  |  |  |
| --- | --- | --- |
|  | Archivo “small” | Archivo “large” |
| Carga de Datos |  |  |
| Req. 1 |  |  |
| Req. 2 |  |  |
| Req. 3 |  |  |
| Req. 4 |  |  |
| Req. 5 |  |  |
| Req. 6 |  |  |

Tiempo de carga en segundos de los requerimientos en la máquina 1.

# **Maquina 2 Resultados**

|  |  |  |
| --- | --- | --- |
|  | Archivo “small” | Archivo “large” |
| Carga de Datos |  |  |
| Req. 1 |  |  |
| Req. 2 |  |  |
| Req. 3 |  |  |
| Req. 4 |  |  |
| Req. 5 |  |  |
| Req. 6 |  |  |

Tiempo de carga en segundos de los requerimientos en la máquina 1.

.

# **Preguntas de análisis**

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

Sí, cómo se mostró en clase los algoritmos de menor complejidad temporal mostraron una rapidez mayor en las dos muestras, y como se predecía la diferencias entre los algoritmos más rápidos con los más lento era mayor con muestras de mayor tamaño.

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

Sí. La diferencia es muy clara, sobretodo con respecto al procesador. En algunos casos los tiempos se duplican de una máquina a otra y ejecutando el mismo proceso.

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

Principalmente a los procesadores. La diferencia en cuanto a las operaciones que puede realizar un computador en un tiempo determinado varía, y por ende la rapidez con la que se ejecuta algoritmo varía también.

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

Los resultados dejan muy claro que para los algoritmos de ordenamiento que usamos son considerablemente mejores cuando se implementan a través de un TAD lista de tipo arreglo.

1. Teniendo en cuenta las pruebas de tiempo de ejecución por todos los algoritmos de ordenamiento estudiados (iterativos y recursivos), proponga un ranking de los mismo de mayor eficiencia a menor eficiencia en tiempo para ordenar la mayor cantidad de obras de arte.
2. Merge.
3. Shell. (aunque shell y quick estabán muy cerca para los datos que usamos fue mejor Shell, probablemente porque había varias fechas iguales.
4. Quick
5. Insertion.