Documento de Análisis

Nombres, Código

En el siguiente documento, registraremos la complejidad, el tiempo de carga y el uso de memoria de cada uno de los requerimientos del Reto 2. Para registrar el uso de memoria utilizaremos la extensión Resource Monitor de VS Code.

***Cargar información del Catalogo***

**Tiempo de carga**

|  |  |
| --- | --- |
| ***Tamaño del archivo*** | ***Tiempo de Ejecución [ms]*** |
| **Small** | 187.5 |
| **5%** | 1328 |
| **10%** | 2250.0 |
| **20%** | 4093 |
| **30%** | 5687.5 |
| **50%** | 8875 |
| **80%** | 14093.75 |
| **Large** | 16125 |

***Req 1.***

**Tiempo de carga**

|  |  |
| --- | --- |
| ***Tamaño del archivo*** | ***Tiempo de Ejecución [ms]*** |
| **Small** | 15.625 |
| **5%** | 15.625 |
| **10%** | 15.625 |
| **20%** | 15.625 |
| **30%** | 15.625 |
| **50%** | 15.625 |
| **80%** | 31.25 |
| **Large** | 31.25 |

**Complejidad**

**Uso de Memoria**

***Req 2.***

**Tiempo de carga:**

|  |  |
| --- | --- |
| ***Tamaño del archivo*** | ***Tiempo de Ejecución [ms]*** |
| **Small** | 46.875 |
| **5%** | 187.5 |
| **10%** | 234.375 |
| **20%** | 343.75 |
| **30%** | 546.87 |
| **50%** | 671.87 |
| **80%** | 687.5 |
| **Large** | 718.75 |

**Complejidad**

**Uso de Memoria**

***Req 3.***

**Tiempo de carga:**

|  |  |
| --- | --- |
| ***Tamaño del archivo*** | ***Tiempo de Ejecución [ms]*** |
| **Small** | 0.0 |
| **5%** | 0.0 |
| **10%** | 15.625 |
| **20%** | 15.625 |
| **30%** | 15.625 |
| **50%** | 15.625 |
| **80%** | 15.625 |
| **Large** | 15.625 |

**Complejidad**

Para el requerimiento se usaron 2 funciones: la primera se encarga de buscar el nombre que ingresa por parámetro en un map cuya pareja (llave, valor) es , esta búsqueda se hace con el método get() el cual tiene una complejidad O(1), se saca su id y se busca en otro map y retorna su valor (el TAD map con los medios), esta función tiene una complejidad temporal total de O(1); La segunda función recorre las llaves del mapa retornado en la primera función, O(#de llaves), usando el método keySet() el cual tiene una complejidad de O(#de llaves), y por cada llave se calcula el número de obras asociadas y se suman a un contador y a la par se calcula cual es el medio(llave) con mayor número de obras asociadas, la complejidad de esta función es de O(#de llaves). Haciendo la suma de complejidades, O (1 + #de llaves), la complejidad del requerimiento es O(#de llaves) donde el #de llaves es el número de medios utilizados por el artista que se quiere consultar.

**Uso de Memoria**

***Req 4.***

Para el requerimiento 4 se usó solo una función: Esta se encarga de recorrer las llaves del índice de nacionalidad, usando el método keySet () que tiene una complejidad O (#de llaves). Se itera por las llaves del mapa, luego se utiliza get () de complejidad O (1) para obtener las nacionalidades. Posteriormente si la nacionalidad existe, utiliza la función getValue de complejidad O (1) para obtener el valor de las obras de la nacionalidad. Luego se agregan a una lista, la longitud de la nacionalidad utilizando size(value) de complejidad O (1), la nacionalidad(key) y las obras de la nacionalidad(value). Luego se organiza la lista por la longitud, utilizando Merge Sort que tiene una complejidad O(n log(n)). Se crea una sublista de la lista y se toma desde la posición 1 a la 10. A esa sublista, se le aplica el método getElement () de complejidad O (1) para obtener las obras de la nacionalidad de mayor longitud. Por último, a la variable donde se almacenan las obras de la nacionalidad de mayor longitud se le hace getElement (), para devolver los tres primeros y tres últimas obras. En conclusión, la complejidad del requerimiento seria O (1 + #de llaves + (n log(n)).

**Tiempo de carga:**

|  |  |
| --- | --- |
| ***Tamaño del archivo*** | ***Tiempo de Ejecución [ms]*** |
| **Small** | 0.0 o 15.625 |
| **5%** | 0.0 o 15.625 |
| **10%** | 0.0 o 15.625 |
| **20%** | 0.0 o 15.625 |
| **30%** | 0.0 o 15.625 |
| **50%** | 0.0 o 15.625 |
| **80%** | 0.0 o 15.625 |
| **Large** | 0.0 o 15.625 |

**Complejidad**

**Uso de Memoria**

***Req 5.***

**Tiempo de carga:**

|  |  |
| --- | --- |
| ***Tamaño del archivo*** | ***Tiempo de Ejecución [ms]*** |
| **Small** | 31.25 |
| **5%** | 203.125 |
| **10%** | 421.875 |
| **20%** | 890.625 |
| **30%** | 1468.75 |
| **50%** | 2421.875 |
| **80%** | 4031.25 |
| **Large** | 5281.25 |

**Complejidad**

**Uso de Memoria**