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**

**Diagrama**

Para visualizar el diagrama: [Diagrama](https://viewer.diagrams.net/?tags=%7B%7D&highlight=0000ff&edit=_blank&layers=1&nav=1&title=Untitled%20Diagram.drawio#R7V1bk5s4Fv41rnmaKSRu5rG7nWRTNUmlKrs7k0fa0DYTDB2M093761dghI0kYgxIgu6TVCVGxuJyzvl07lqYd7vnD5n%2FuP2UBmG8wEbwvDBXC4yRYznkv2Lk5TiyXOLjwCaLguqk08DX6H9hNWhUo4coCPeNE%2FM0jfPosTm4TpMkXOeNMT%2FL0qfmaQ9p3Lzqo78JuYGvaz%2FmR%2F%2BKgnxbjSLHO33xrzDabKtLL7F7%2FOLeX3%2FfZOkhqa63wOa78s%2Fx651P56oedL%2F1g%2FTpbMh8tzDvsjTNj592z3dhXLxb%2BtqOv3vf8m1931mY5F1%2BkDyiz8ntf7cfnvxvibX2P%2FzHffm9otVPPz6E9DGcmMx3%2B0g%2BbIoPdOAhJdchj5G%2FVK%2FO%2BXFI6Re%2F70vC3pATkPv4fPqyIIm%2Fbv7gLj1kUZiRrz6HT%2BenHi945%2Bd%2BnG7odckTHS%2FdvB0yfHaLuHFn%2BGkb5eHXx%2BOFnwjjkpO2%2BS4mR4h8LKkWBtXRQxTHd2mcZuVvzff2za1tkPF9nqXfw7NvPKP4Wz3x2Ti6Kf7W9%2FEzzPLwuZVKqKY9kakw3YV59kJOqX5gVtxSSZPpVMdP57xZjW3P2NKqxvxKHDb1zCeWIB8qrriCQywBhzBvO0yCm0ISyVGSJmHzXRfv6r2%2Fi%2BLiiZqUP35ZQQJyyXH4HOV%2FVz8sPn8jn40%2F7OpoVbwjgx68VAccmdASvysesJVMsX8fxre1%2BJ6dUgkwpWQYcNjB0JG8B%2FJIFX%2F%2FQsR4ep8R1BbQk45lYezn0c%2FmbYiIXF3hSxqVYlqxk42a%2FGSxfHK8%2FepX5%2BjBTIQNhjGXzES5n23CnJuo5Ln6sfuzoa2SDbWylC5WYSnseiOxCstykjnFuYpT1rG%2F30drYJZhzNIbVhytvOICr6jnFYRHYhZuIsncsgRu0cAtnveHPRK%2F8FNJ5hgPOEY9x%2BDxOEYwlWSOofo6sIxa9WU0%2FUXtkoQQsIsGdlmOhjCCqWSzDAaWUc8y9ngsI5hKNstc530BlhmFZdwBLGNfnEoyy2Cb5xDy5r9Wh2mWb9NNmvjxu9Po7Wn0zzR9rBjnnzDPXyrm8A95Ole26uAqRo6Y%2FwYyloWZNUcxeog8copjSjdZ%2FpRm3%2FetQSWGV%2BcbQOJMmIrY2gJI2AEkuB4JXCVIgJBiKBA5XCHePCG48LTDhQtwcT1cLJXABVbt3RB53HXDBVElon3%2B2d%2BFtTZxnwnQ4g2DCKLJAecogpWiyBJQ5HoU8eSgCJvTpBpFRFGYaaDIX%2Bc2CaAIs9qIUMRUiiIeoMj1%2BW6GmMoj6yKmZylFEXr5SaHIKiQUyneERoAiLShiupZmXYRCFqDINSjSQuXBKIKpI5w6Q9k0R9k4IorY6saR23ATJSs%2FB5PmBCMco7i6jRqEpqaPmHNAEnw88SjXvzjRGwo5w3RNPEFc%2BEwevOCcKH8BYBDrF7Z2WDDxxFBhFvqFqcRKcVRrF%2BYEUeTo6%2FgYAISIIcQRQYjSoAuaHIbMQrOwumoWQ8FmmGYhKuHUjQmlsXET%2FDhEWQjA0Ja2JQIGS6luAWlc1%2BOCaUnRLWy9aVwmDyN%2FkmWd4w%2FC%2FnmTuMR%2B2CTk85q8SUJW87YQkmjtxzfVF7soCErOyUICMf59OVUha4%2FFw5SPZ98u7FUxF2GdfcULHCtUZb%2FnMk2HGBb6Jb8xCCFP3jmfNi%2FujkDaTWnSDrkXPaRdTtKmw9YcKJb2DmXUFxbaq6g84VXZ5uLXnJSK1mRpUkovJltKyevJXv6ukLg8OGvHURye%2BnGUR7Qhx%2BuS7mVHHd8cmjwxDLk71Ca%2FEXFlgVO7uCoyrUFcayns)