OBSERVACIONES DEL LA PRACTICA

Javier Cerino Cod 202020873

Marco Zuliani Cod 202022412

1. ¿Qué estructura de datos se usa para este índice?

Para el índice de Authors se utiliza una estructura de datos abstracta de tipo MAP. El tipo de tabla se hash que se implemente es un “Chaining”.

1. ¿Cuántos elementos se espera almacenar inicialmente?

Se espera inicialmente cargar 800 autores.

1. ¿Cuál es el factor de carga?

El factor de carga de este map es de 4.0.

1. ¿Con cuántos elementos serán necesarios agregar para hacer re-hash de la tabla?

Para hacer un re-hash de la tabla como máximo se pueden almacenar 844 elementos, esto lo sabemos gracias a la siguiente ecuación donde es el factor de carga que en este caso corresponde a 4. N es el número de elementos a guardar en la tabla de Hash. M es el tamaño de la tabla que se calcula con la función nextPrime(numelements//loadfactor) que en este caso el numelements es de 800, siguiendo la ecuación el siguiente primo sería 211, por lo que M asumiría este valor.

1. ¿Qué hace la instrucción “**mp.put(...)”**?

La instrucción “mp.put( … )” llama a la función put del map. Esta función tiene como propósito agregar un elemento a la tabla de hash en cuestión. Esta función varia dependiendo del tipo de tabla de hash que se utilice. En este caso la tabla de hash utilizada para esta instrucción es de tipo “chaining”. En adición en caso de que supere el límite permitido por el load factor se realizaría un re-hash de la tabla.

1. ¿Qué papel cumple **“book[‘goodreads\_book\_id’]”** en esa instrucción?

El papel que cumple “book[‘goodreads\_book\_id’]” en esta instrucción es el de indicar la llave de la pareja llave-valor. En este caso siendo una tabla de hash de tipo chaining esta llave se utiliza para buscar el bucket donde debería estar almacenado este valor. En caso de ya existir una pareja llave valor con esta misma llave se remplaza el valor anterior con el nuevo ingresado por esta función. En caso de que este valor no haya sido agregado con anterioridad al bucket de la tabla de hash, se agrega a este. Una vez realizado este procedimiento se recalcula el load factor y se compara con el de la tabla de hash, en caso de que este nuevo load factor supere al default de la tabla se realiza una operación de re-hash.

1. ¿Qué papel cumple **“*book”*** en esa instrucción?

“book” en esta instrucción será el valor de la pareja llave valor. En caso de ya encontrar una llave que coincida con la llave de esta pareja, se reemplazará el valor existente con este nuevo valor.

1. ¿Qué hace la instrucción **“mp.get(…)”**?

La instrucción “mp.get( …)” como su nombre lo indica, retorna un valor. Este valor corresponde a la pareja llave-valor de una llave específica. En este caso, la llave que se busca es el year que ingresa como parámetro. Una vez analizada la tabla de hash se procede a encontrar la posición de la llave que almacena los elementos. En caso de encontrarla retorna la pareja llave-valor. En caso de no existir retorna None:

1. ¿Qué papel cumple **“*year”*** en esa instrucción?

Year en esta instrucción es la llave que se ingresa como parámetro. Una vez realizada la función mp.get asumirá la pareja de llave-valor o en el caso de no existir esta pareja asumirá el valor de None.

1. ¿Qué hace la instrucción **“me.getValue(…)”**?

La función “me.getValue( …)” devuelve el valor de la pareja llave valor, en este caso retorna los valores de la pareja year. Al final, en este caso solo retorna los [‘books’] de este valor.