OBSERVACIONES DEL LA PRACTICA

Lina Maria Gomez 201923531

Juan Sebastian Montealegre Cod 202012723

1. ¿Qué estructura de datos se usa para este índice?
   1. Para este índice el cual es “authors” autores de libro se implementa un map, el tipo de tabla hash sobre la que se implementa es “Separate Chaining ”. Esta se utilizó ya que es una estructura que guarda parejas llave-valor en la que no se repite el valor llave (autor) y crea una linked-list para la cantidad total de los autores. Es preferible utilizar para este caso un map ya que el tiempo de búsqueda se puede reducir a O(1) a diferencia de otras EDA.
2. ¿Cuántos elementos se espera almacenar inicialmente?
   1. Se espera que inicialmente se almacenen 800 elementos (autores), esto es una cantidad que nosotros elegimos, esta cantidad puede aumentos en caso de hacerse un rehash.
3. ¿Cuál es el factor de carga?
   1. El factor de carga es de 4.0, este factor de cargada es escogido por nosotros, este factor de carga nos indica que en general van a ver 4 datos en el bucket por cada posición de nuestro map. Este factor de 4,0 es el recomendado para un buen rendimiento para este tipo de tabla de hash.
4. ¿Con cuántos elementos serán necesarios agregar para hacer re-hash de la tabla?
   1. Los elementos que van a ser necesarios para hacer un rehash, es cuando nuestro current factor sea mayor o igual al rehash, como consecuencia se van a tener que asignar de nuevo todos los elementos lo cual es muy costoso en tiempo, también esto generara que se use más espacio en memoria.
5. ¿Qué hace la instrucción “**mp.put(...)”**?
   1. La función put ingresa una pareja llave valor, a nuestra tabla de hash. Esta función toma como argumentos tres datos, el primero el map en donde lo va a guardar, el segundo el key y el tercer el value, para nuestro caso el map va a ser el de book[“bookIds”] nuestro key va ser el id, y el valor va  a ser book.
6. ¿Qué papel cumple **“book[‘goodreads\_book\_id’]”** en esa instrucción?
   1. Para este caso va ser nuestro segundo argumento de nuestra función put, el cual es el key que va a llevar un valor respectivo.
7. ¿Qué papel cumple **“*book”*** en esa instrucción?
   1. EL papel que cumple el tercer parámetro“book”en esa instrucción, es el valor que va a tomar nuestra llave la cual es el segundo parámetro de esta manera el segundo y tercer parámetro significan la llave valor que va a entrar a nuestro map , el cual es el primer argumento.
8. ¿Qué hace la instrucción **“mp.get(…)”**?
   1. La función mp.get() retorna una pareja <llave-valor>, donde la llave es igual a key entregada por parámetro. En el caso de que la llave no exista la función se encarga de retornar None. Además, esta tiene dos entradas por parámetro: el el map en el que se va a buscar y la llave. Extrapolando esta información a la hoja de trabajo, en la primera posición se va a buscar en catalog[“years”] y en la segunda el año, year, a buscar.
9. ¿Qué papel cumple **“*year”*** en esa instrucción?
   1. Tal como se indicó en el punto anterior el papel de year es la llave, key, asociada a la pareja de libros que fueron publicados en ese año particular.
10. ¿Qué hace la instrucción **“me.getValue(…)”**?
    1. Me está asociado a la librería mapentry que se encarga de recibir una entrada <llave, valor> de un map particular. Solo tiene un parámetro que es entry. En este, caso la función. getvalue recibe de entrada year que es la pareja <year, libro> y retorna solamente el value de dicha llave; es decir, cada libro encontrado para un año x.