

OBSERVACIONES DE LA PRACTICA

Tomás Ángel Gallón Cod 202020366

Daniel Osorio Cárdenas Cod 202022996

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

Para este índice se utiliza una estructura de datos de tipo chaining.

2) ¿Cuántos elementos se espera almacenar inicialmente?

Inicialmente se espera guardar 800 autores.

3) ¿Cuál es el factor de carga?

El factor de carga es de 4.0

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

Si tenemos un tamaño inicial de tabla de 800 autores, y un factor de carga de 4.0, entonces para que sea necesario hacer un rehash se necesitaría que los datos almacenados en la tabla sea superior al tamaño inicial de la tabla multiplicado por el factor de carga. Es decir, los datos almacenados en la tabla deberán ser iguales o superiores a 3200 autores para hacer re-hash de la tabla. Como inicialmente teníamos 800 autores, se necesitarían agregar más de 2400 autores para que la tabla inicie el rehash

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

Ingresa una pareja llave,valor a la tabla de hash. Si la llave ya existe en la tabla, se reemplaza el valor. Esta instrucción recibe el map donde se va a guardar la pareja, la llave y el valor correspondiente a la respectiva llave. Para el caso de la función del model “addBook”, el map que recibe es el *catalog[‘bookIds’]*, de llave recibe el id correspondiente del libro -es decir- *catalog[‘goodreads_book_id’]*, y por ultimo el valor a almacenar es el mismo *book*.

6) ¿Qué papel cumple “book[‘goodreads_book_id’]” en esa instrucción?

El *book[‘goodreads_book_id’]* cumple la función de obtener el id del libro que se quiere agregar para poder utilizarlo como llave para referenciar al libro y agregarlo al *catalog[‘bookIds’]*, que es donde se guardan todos los ids de los libros y las referencias a los mismos.

7) ¿Qué papel cumple “book” en esa instrucción?

El *book* en esa instrucción refiere al libro que se quiere almacenar en el map de los bookIds.

8) ¿Qué hace la instrucción “mp.get(...)”?

El *mp.get* retorna la pareja llave, valor, cuya llave sea igual a key. Esta función recibe como parametros el map y la llave a buscar. Para el caso de la función *getBooksByYear*, el map donde se va a buscar es el *catalog[‘years’]*, y la llave que se va a buscar en ese map es el año (*year*).

9) ¿Qué papel cumple “year” en esa instrucción?

Year es un parámetro de la función *getBooksByYear* en donde entra el año que se desea buscar, y esa variable es la misma que se le ingresa al *mp.get()* como llave a buscar dentro del map *catalog[‘years’]*. En otras palabras *year* es el año que se busca dentro del map de todos los años.

10) ¿Qué hace la instrucción “me.getValue(...)”?

El *me.getValue()* retorna el valor de la pareja del entry. Para la función *getBooksByYear*, lo que hace es que si encuentra el año que se está buscando, se le hace un *me.getValue()* a ese año para obtener la pareja llave valor. Esta pareja llave valor es un diccionario con las llaves *key* y *value*. Entonces, con la función *me.getValue* se saca solamente la información de ese año. La información de ese año es un diccionario con las llaves ‘year’ y ‘book’, en la primera se encuentra el año y en la segunda se almacena una lista con todos los libros correspondientes a ese año. Por eso es que al final de la instrucción *me.getValue(year)* sacan la llave [‘books’], para así obtener únicamente la lista de libros para ese año específico. La instrucción completa es *me.getValue(year)[‘books’]*.