OBSERVACIONES DEL LA PRÁCTICA

Juan José Osorio 202021720

Thais Tamaio 202022213

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

La estructura de datos que se utiliza para almacenar la información correspondiente de los diferentes autores es un mapa (o tabla de símbolos) de tipo “Chaining”. En esta estrucutra de datos, se almacena la información por medio de las parejas <llave, valor>. Este tipo de tabla de símbolos en específico es conocida como “Separate Chaining”, ya que en caso de alguna colisión a la hora de guardar toda la información de los autores en el mapa, se guardan todas las parejas <llave, valor> (que coinciden en una misma posición) en una lista denominada bucket.

1. ¿Cuántos elementos se espera almacenar inicialmente?

Según la declaración de la llave “authors”, se puede apreciar (que de manera inicial) se espera almacenar 800 datos. Esto es evidente gracias al parámetro “numelements”, el cual representa el tamaño incial de la tabla. Este parámetro tiene un valor por defecto de 17 elementos. No obstante, al declarar la llave “authors”, se estbalece un valor de 800 elementos iniciales para la tabla.

1. ¿Cuál es el factor de carga?

Según la declaración de la llave “authors”, se puede apreciar que el factor de carga que tendrá el nuevo mapa es de 4.0. Esto es evidente gracias al parámetro “loadfactor” el cual representa el factor de cargar inicial de la tabla. Este parámetro tiene un valor por defecto de 0.5. No obstante, al declarar la llave “authors”, se establece un factor de carga con un valor de 4.0.

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

Tomando como referencia los valores que entran por parámetro al declarar la llave “authors”, los cuales son: Un factor de carga de 4.0 y un tamaño inicial de la tabla de 800 datos. Para determinar la cantidad de elementos que se necesitan para hacer re-hash de la tabla se utiliza la siguiente fórmula, donde es el factor de carga, es el número de elementos que se espera almacenar inicialmente y es la cantidad de elementos necesaria para hacer re-hash:

Al reemplazar los valores que se determinan por medio de la declaración de la llave “authors” se obtiene:

Por lo que se puede concluir que se necesitan agregar 3200 elementos para hacer el re-hash de la tabla.

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

Dentro de la función addBook, la instrucción m.put() se encarga de manera general, de agregar parejas de <llave, valor> a un mapa, o en caso de que ya exista esa llave dentro de la tabla, reemplaza el valor de la llave existente por el nuevo. Esta función recibe como parámetros: el mapa en el que se desea guadrar la pareja, la llave y el valor. De manera específica, la instrucción m.put en la función “addBook” del modelo se encarga de agregar en el mapa “catalog[“booksIds”]” el ID de cada libro como llave, y la información de cada libro respectívamente como valor.

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

En la instrucción mp.put, “book[‘goodreads\_book\_id’]” corresponde a la llave que se le desea agregar al mapa “catalog[“booksIds”]”. En este caso, la llave es el ID del libro que desea agregarse en el catálogo.

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

En la instrucción mp.put, “book” corresponde al valor que desea agregarse el mapa “catalog[“booksIds”]” bajo la llave **“book[‘goodreads\_book\_id’]”**. En este caso, el valor correponde a toda la infromación de un libro en específico. En book se guarda una lista de tuplas con la información de un libro en específico. La ***información*** que se almacena por cada libro es la siguiente: 'book\_id', 'goodreads\_book\_id', 'best\_book\_id', 'work\_id', 'books\_count', 'isbn', 'isbn13', 'authors', 'original\_publication\_year', 'original\_title', 'title', 'language\_code', 'average\_rating', 'ratings\_count', 'work\_ratings\_count', 'work\_text\_reviews\_count', 'ratings\_1', 'ratings\_2', 'ratings\_3', 'ratings\_4', 'ratings\_5', 'image\_url' y 'small\_image\_url'.

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

De manera general, la instrucción mp.get se encarga de retornar el valor de acuerdo a una llave indicada. La instrucción mp.get dentro de la función “getBooksByYear”, recibe como parámetros el mapa “catalog[‘years’]” y la llave asociada al valor que se desea encontrar (“year”), es decir al año con el que se desea realizar la búsqueda. Esta intrucción retorna una pareja <llave, valor>, es decir el año ingresado por parámetro y los libros (con su respectiva ***información***, la cual es igual a la mencionada en el punto 7) que fueron publicados en dicho año respectívamente.

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

La variable “year” que es el parámetro “key” en la instrucción mp.get establece el nombre de una llave en específico dentro del mapa llamado “catalog[‘years’]”, para así, poder encontrar todos los libros asociados al año ingresado por el usuario.

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

De manera general, la instrucción me.getValue recibe como parámetro una pareja <llave, valor> y retorna únicamente la llave de la pareja. En la función “getBooksByYear” la intrucción me.getValue se ingresan como parámetro las parejas de <llave, valor> (que se obtuvieron por medio de mp.get), las cuales están guardadas dentro de una variable llamada “year”. En estas parejas, la llave es el año ingresado por el usuario y el valor son listas de tuplas con toda la infromación de cada libro que se publicó en ese año en específico (igual a la ***información*** mencionada en el punto 7). Como me.getValue retorna únicamente el valor de una pareja de <llave, valor>, se retorna en este caso en específico la información de cada libro que se publicó en el año que se encuentra en esa llave.