OBSERVACIONES DEL LA PRACTICA

Daniel R Alonso A Cod 201419873

Nicolás Díaz Montaña Cod 202021006

catalog['authors'] = mp.newMap(800,

maptype='CHAINING',

loadfactor=4.0,

comparefunction=compareAuthorsByName)

Figura 1 – Declaración de la llave “Authors”

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

Se esta utilizando la estructura de datos Map utilizando el tipo de tabla hash “separate chaining” como se puede apreciar en la Figura 1.

1. ¿Cuántos elementos se espera almacenar inicialmente?

Como se puede apreciar en la Figura 1, el parametro de numelements (numero de elementos) de newMap() es de 800 lo cual significa que se espera guardar incialmente esa cantidad de elementos.

1. ¿Cuál es el factor de carga máximo?

Según el parametro loadfactor (factor de carga) de newMap(), como se evidencia en la Figura 1, es de 4.0.

-------------------------------------------------------------------------------------------------------------------------------

def put(map, key, value):

""" Ingresa una pareja llave,valor a la tabla de hash.

Si la llave ya existe en la tabla, se reemplaza el valor

Args:

map: El map a donde se guarda la pareja

key: la llave asociada a la pareja

value: el valor asociado a la pareja

Returns:

El map

Raises:

Exception

"""

return ht.put(map, key, value)

Figura 2 – Función put de MAP.

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

Es la que se encarga de meter un nuevo elemento con su respectiva llave en un diccionario utilizando los parametros ingresados por el usuario. Estos parametros son map (el diccionario al cual se le va agregar el elemento), key (la llave asociada al elemento) y value (el elemento en si).

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

**“book[‘goodreads\_book\_id’]” cumple el papel de la llave del elemento que se va ingresar al diccionario “catalog[‘bookIDs’]”.**

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

“book” es el elemento que se va ingresar al diccionario “**catalog[‘bookIDs’]”, y esta asociado a la llave**

**“book[‘goodreads\_book\_id’]”.**

-------------------------------------------------------------------------------------------------------------------------------

def getBooksByYear(catalog, year):

"""

Retorna los libros publicados en un año

"""

year = mp.get(catalog['years'], year)

if year:

return me.getValue(year)['books']

return None

Figura 3 – Función getBooksByYear()

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

La función de map get() se encarga de devolver la pareja llave, valor de un diccionario dado, cuya llave sea igual a la llave ingresada por el usuario. En este caso se esta pidiendo encontrar el valor de la llave “year” del diccionario “catalog[‘years’]”.

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

Si hablamos “year” tipo parámetro es la llave ingresada por el usuario o que se esta pidiendo. Si hablamos de la variable es una pareja <<llave, valor>> que en este caso es el valor asociado al parametro key <<year>>.

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

Retorna el valor de una pareja de un Map, en este caso la pareja en cuestion es la varible “year”.