

OBSERVACIONES DEL LA PRACTICA

1. María Paula Nizo Vega, m.nizo@uniandes.edu.co, 202213902

2. Paul Paffen Suarez, p.paffen@uniandes.edu.co, 202222496

3. Andres Camilo Caballero Ayala, Ac.caballero@uniandes.edu.co, 202216295

- 1) ¿Qué estructura de datos se usa para este índice?
- 2) ¿Cuántos elementos se espera almacenar inicialmente?
- 3) ¿Cuál es el tamaño de las tablas de hash para **'years'** y **'authors'**?
- 4) ¿Cuál es el factor de carga máximo?
- 5) ¿Qué hace la instrucción **"mp.put(...)"**?
- 6) ¿Qué papel cumple **"book['goodreads_book_id']"** en esa instrucción?
- 7) ¿Qué papel cumple el tercer parámetro **"book"** en esa instrucción?
- 8) ¿Qué hace la instrucción **"mp.get(...)"**?
- 9) ¿Qué papel cumple **"year"** en esa instrucción?
- 10) ¿Qué hace la instrucción titulada **"me.getValue(...)"**?
- 11) ¿Cuál es la llave y el valor del nuevo índice implementado?, describa el tipo de dato y/o las estructuras de datos involucradas.
- 12) ¿Qué pasa cuando existan dos o más libros con el mismo título (ej.: diferente edición)?
- 13) Con el índice implementado ¿Cómo solucionaría que dos o más libros tengan el mismo título?

- 1) Tablas de hash, específicamente con el tipo PROBING y CHAINING
- 2) Para "years" se espera almacenar inicialmente 40 elementos y para "authors" 800 elementos
- 3) El tamaño de las tablas de hash para 'years' es 80 y para 'authors' es 200.
- 4) El factor de carga puede depender del tipo de tabla de hash usada, para el caso de los PROBING se usa generalmente un valor menor a 1 y para CHAINING un valor mayor (Estos factores de carga se deberán a las colisiones). En el caso del código mostrado el factor de carga máximo usado es 4.
- 5) Ingresa una pareja llave, valor a la tabla de hash de interés ("PROBING" o "CHAINING") y si el valor ya existe en la tabla, se reemplaza. Teniendo como parámetros, la tabla, la llave y el valor de asociado a la pareja.
- 6) Es la llave con la que se guardará el valor, que en este caso será el id de cada libro y se asociará a un valor.
- 7) Es el valor que se desea asociar a la llave del id de cada libro, en este caso será toda la información de cada libro que puede ser un diccionario con toda su información.
- 8) Retorna una pareja llave valor de una tabla de hash, recibiendo como parámetro la tabla y la llave que se desea extraer.
- 9) Year es una variable donde se almacena una pareja llave, valor y que en caso de no encontrar funcionará como un valor que convierte en falsa la condición if year, devolviendo None.
- 10) Extrae el valor de una pareja llave, valor dada. Y recibe de parámetro únicamente la pareja.
- 11) La llave viene siendo el título del libro (En este caso "original_title") y el valor es la información del libro o libros correspondientes a ese título, que sería una lista de diccionarios con todos los

libros o un único diccionario. Y estas parejas están asociadas desde catalog["titles"] que contiene las llaves.

- 12) Si existen más de dos libros con el mismo título, es muy posible que solo devuelva uno de ellos, entonces esto genera que se tenga que modificar el código. Esto puede ser causado por múltiples factores del Libro, género, tema, trama, edición y autor completamente distintos, pero con igual título.
- 13) Se crea un Array o una lista enlazada para el valor si no existe el título anteriormente, y en la que se guarde después cada libro si se repite el título y luego se muestren en pantalla al devolverse todos los libros juntos en el Array como en el caso del código.

```
if mp.contains(catalog['titles'], title['original_title']):
    book = mp.get(catalog['titles'], title['original_title'])
    lista = me.getValue(book)
    lt.addLast(lista, title)
else:
    lista = lt.newList("ARRAY_LIST")
    lt.addLast(lista, title)
    mp.put(catalog['titles'], title['original_title'], lista)
```