

Isai Daniel Chacón Silva 201912015

Nicolás Aparicio Claros 201911357

Estructuras de Datos y Algoritmos

Universidad de los Andes

LabMaps

Preguntas

- ¿Qué estructura de datos se usa para este índice?
- ¿Cuántos elementos se espera almacenar inicialmente?
- ¿Cuál es el factor de carga máximo?

```
"""
Este índice crea un map cuya llave es el autor del libro
"""
catalog['authors'] = mp.newMap(800,
                              maptype='CHAINING',
                              loadfactor=4.0,
                              comparefunction=compareAuthorsByName)
```

Así como se observa en la sección de código, la estructura de datos utilizada para los autores es un mapa de tipo 'Chaining', cuyos valores esperados a almacenar inicialmente son 800, con un factor de carga máximo de magnitud 4.

- ¿Qué hace la instrucción "mp.put(...)"?
- ¿Qué papel cumple "book['goodreads_book_id']" en esa instrucción?
- ¿Qué papel cumple el tercer parámetro "book" en esa instrucción?

```
mp.put(catalog['bookIds'], book['goodreads_book_id'], book)
```

La instrucción mp.put es la implementación del TAD map que se encarga de anexar una pareja llave-valor a la tabla de hash especificada por parámetro, en este caso, el catálogo de bookIds. Así mismo, el parámetro book['goodreads_book_id'] indica la llave del mapa. Específicamente, 'goodreads_book_id' es una de las columnas del archivo .csv que contiene a los id's. El value para cada id vendrá dado por el libro en sí, que es representado realmente por cada uno de los valores asociados a las columnas del archivo .csv con toda la información de los libros.

- ¿Qué hace la instrucción "mp.get(...)"?
- ¿Qué papel cumple "year" en esa instrucción?
- ¿Qué hace la instrucción "me.getValue(...)"?

```
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
```

La instrucción de `mp.get` es la implementación del TAD mapa, para el cual se devuelve el par key-value, en este caso el año y el value que es un diccionario de Python. Este diccionario posee a su vez como key el año de interés y como value una lista (cuya implementación nativa es simplemente encadenada) con todos los libros encontrados en el catálogo 'years', asociados al 'year' ingresado por el usuario. Esta búsqueda se realiza sobre el mapa del catálogo de años.

Por su parte, la función `me.getValue` se encarga de devolver solo el value asociado al diccionario `year["value"]`, para el cual value fue declarado como variable dos líneas antes de la ejecución de este comando. Esto quiere decir que el método retornará solamente la lista encadenada con los libros previamente encontrados (de manera análoga `year["value"]["books"]`).