

OBSERVACIONES DEL LA PRACTICA

Estudiante 1 ---- Ehimar Andres Vargas Malaver, e.vargasm@uniandes.edu.co, 202014902.

Estudiante 2 Cod XXXX

Preguntas de análisis

- 1) ¿Qué relación encuentra entre el número de elementos en el árbol y la altura del árbol?

Debido a que el map es un BST es “imposible” saber si el mapa organizo la información de manera correcta o siquiera saber si el mapa ordenado se encuentra balanceado, por lo tanto es imposible crear una relación entre los elementos de árbol y su altura.

- 2) ¿Si tuviera que responder esa misma consulta y la información estuviera en tablas de hash y no en un BST, cree que el tiempo de respuesta sería mayor o menor? ¿Por qué?

El tiempo de respuesta depende de la manera en cómo se organicen los datos dentro de la tabla de hash, si se organizan por años solo sería buscar en ese año y retornar el tamaño de la lista dentro de esa llave, y luego en la del año siguiente, por lo tanto el tiempo de respuesta sería $O(n)$, si los datos dentro de la tabla se organiza de otra manera es probable que la complejidad de consultar la información seria mucho mayor, por lo tanto, un BST es muy rápido y la estructura de datos adecuada para almacenar esta información.

- 3) ¿Qué operación del TAD se utiliza para retornar una lista con la información encontrada en un rango de fechas?

Primero se utiliza la función `mo.get()` la cual busca la fecha en la que ocurrió el crimen para así reducir la cantidad de información en la que se debe buscar.

```
193 def getCrimesByRangeCode(analyzer, initialDate, offensecode):
194     """
195     Para una fecha determinada, retorna el numero de crímenes
196     de un tipo específico.
197     """
198     crimedate = om.get(analyzer['dateIndex'], initialDate)
```

```
64 def get(map, key):
65     """
66     Retorna la pareja llave-valor con llave igual a key
67     Args:
68         map: La tabla de símbolos
69         key: La llave asociada a la pareja
70     Returns:
71         La tabla de símbolos con la nueva pareja
72     Raises:
73         Exception
74     """
75     return om.get(map, key)
```

Luego se obtiene el map donde guarda la información que está dentro de un diccionario en la llave ['offenseIndex'], luego, dentro de este map se guarda la información dependiendo del tipo de ofensas, luego se saca el valor de la llave en la ofensa ingresada por parámetro y se retorna el tamaño de la lista que estaba en la llave.

```
● 193 def getCrimesByRangeCode(analyzer, initialDate, offensecode):
194     """
195     Para una fecha determinada, retorna el numero de crímenes
196     de un tipo específico.
197     """
198     crimedate = om.get(analyzer['dateIndex'], initialDate)
199     if crimedate['key'] is not None:
200         offensemap = me.getValue(crimedate)['offenseIndex'] # obtengo
201         numoffenses = m.get(offensemap, offensecode)
202         if numoffenses is not None:
203             return m.size(me.getValue(numoffenses)['lstoffenses'])
204     return 0
205
```

