**OBSERVACIONES DEL LA PRACTICA**

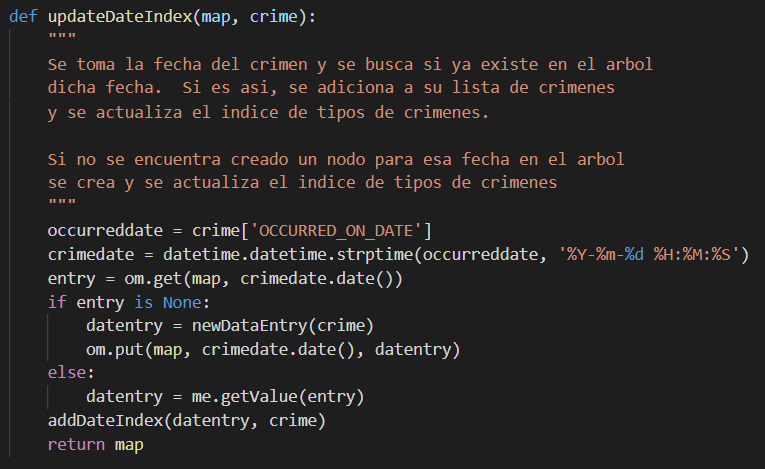
Juan Sebastián Ortega Romero Cod 202021703

Yesid Almanza Cod 201921773

# **Preguntas de análisis**

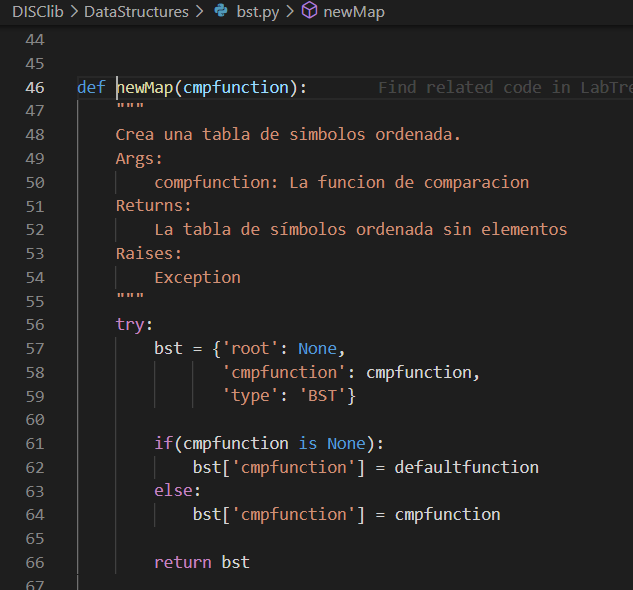
1. ¿Qué relación encuentra entre el número de elementos en el árbol y la altura del árbol?

**Alutra del arbol:** 29 || **Elementos del arbol:** 1177



**Figura 1:** Función updateDateIndex()

La función anterior es utilizada por el algoritmo para añadir crímenes con base en su fecha, de forma que la fecha sirve como la llave a una sub-lista de crímenes (valor). En este caso, es posible observar que el TAD BST convierta a cada una de estas fechas individuales (llaves) en un nodo particular. Por lo tanto, lo primero que es posible asumir es que el número de nodos equivale al número de fechas únicas que se ven representadas dentro de la base de datos.



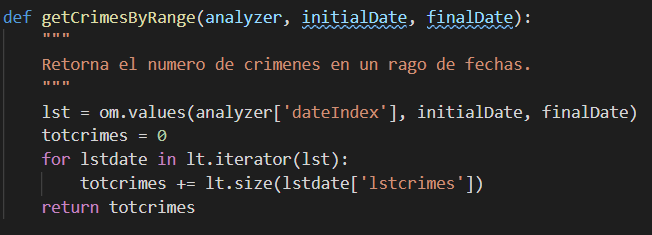
**Figura 2:** Función newMap() en BST.py

Los nodos mencionados anteriormente están siendo añadidos progresivamente a un árbol binario de tipo BST. Como se observa en la figura 2, esta estructura va añadiendo elementos de forma ordenada basándose en un factor de comparación. En este caso el factor de comparación esta determinado por el valor de la fecha de cada nodo (su propia llave) por lo que la altura total del árbol dependerá de que tan diferentes son las fechas de los crímenes. La altura total del árbol de 29 se relaciona con 1177 elementos de forma que estos están repartidos en 29 ramificaciones descendentes. Por lo tanto, cualquiera de estas 1177 fechas puede ser encontrada con tan solo *log2(29)* comparaciones.

1. ¿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 resultaría siendo mayor. A pesar de que una tabla de hash podría acceder a cada llave que representa una fecha de forma relativamente simple (utilizando una complejidad constante de O(K)), el hecho de que las fechas no estén relacionadas entre sí de forma secuencial en una tabla de hash hace que se necesario repetir esta acción con todas las fechas dentro del intervalo de forma individual, lo que inevitablemente acumulará el tiempo de respuesta. Por otro lado, el BST simplemente necesita acceder a la llave que contiene la fecha inicial, obtener su peso y luego acceder a la llave que contiene la fecha final, obtener también su peso y finalmente restar los dos pesos obtenidos. Siempre que la información cargada esté relacionada entre sí, será más sencillo obtener la información respecto a esta por lo que se ahorran múltiples consultas.

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



**Figura 3:** Función getCrimesByRange

Como se observa en la Figura 3, para encontrar una lista con la información encontrada en un rango de fechas determinado se utiliza la función **om.values()** la cual recibe tres parámetros: el catalogo con el mapa organizado, la fecha inicial (llave inicial) y la fecha final (llave final). A partir de esta información retorna una lista de listas, donde cada sub-lista representa a los valores de una llave individual que pertenece al rango de búsqueda.