

Observaciones Laboratorio 7

Integrantes del grupo

Grupo 11: **Federico Melo Barrero.**

202021525.

f.melo@uniandes.edu.co

Juan Camilo Prieto Avella.

201814815.

jc.prietoa@uniandes.edu.co

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

Se evidencia que para un número de 1177 parejas llave valor almacenadas en el árbol binario se tiene una altura de 29. Por un lado, eso indica que la altura del árbol dista sustancialmente del peor caso posible, en el que los datos ingresan de forma ordenada y por ende el árbol binario tiene una única rama y termina siendo en realidad una lista, con altura igual al número de elementos menos uno: 1176. Por otro lado, se ve que el árbol no está perfectamente balanceado, pues tampoco se tiene el mejor caso posible que sería una altura de $\lfloor \log_2 n \rfloor = 10$. Una altura de 29 es relativamente pequeña, lo cual permite que las búsquedas en el árbol binario tengan un comportamiento aproximadamente logarítmico.

Además, existe una relación en la creación de la altura del árbol porque está siempre va a depender de la distribución de las filas de la base de datos y el tamaño de esta porque si nos llegaran a entregar una base de datos ordenada al final tendríamos que el árbol se termina comportando como una lista por la manera que crea la estructura, entonces lo ideal es tener siempre una base que no esté ordenada y el número de elementos del árbol va a afectar siempre la altura por lo que habrán más o menos objetos para organizar en la estructura de datos

b) ¿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 seria mayor o menor? ¿Por qué?

Si se hiciera la misma consulta en una tabla de hash el tiempo de respuesta seria mayor porque esta estructura en particular es muy mala sacando rangos, ya que no está ordenada y no se puede ordenar por lo tanto la única manera de sacar un rango es ir llamando fecha por fecha que este dentro de la estructura que tiene un costo de $O(1)$ pero se repetiría por la cantidad de fechas dentro de este rango. Por el otro lado, un BST es una estructura ordenada entonces se puede buscar de un rango mínimo al máximo y sacar todos los valores que estén entre ambas ubicaciones lo que nos permite tener un algoritmo mucho menos costoso $O(\lfloor \log_2 n \rfloor)$.

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

En el archivo **view.py** se pide al usuario una fecha inicial y una fecha final para determinar el rango. Esas fechas entran como parámetros a la función **getCrimesByRange**, y van a tomar el papel de las llaves para definir el rango de fechas del que se quiere obtener una lista con información. Al interior de la función **getCrimesByRange**, se utiliza la operación **values(om, keylo, keyhi)** del TAD **orderedmap** (mapa ordenado) para retornar una lista con todos los valores del árbol que se encuentran entre dos llaves de la forma **[keylo, keyhi]**, en este caso **[fecha_inicial, fecha_final]**.

Nótese que la función **values(om, keylo, keyhi)** del TAD **orderedmap** no ejecuta ninguna acción de forma directa, sino que redirige a la función de la estructura de datos que se está utilizando, por ejemplo, a la función **values(bst, keylo, keyhi)** de la estructura de datos **bst** (árbol binario, *binary search tree*).