

LAB-08-SEC-09

Estructura de Datos y Algoritmos 2024

Julian Restrepo

j.restrepo112@uniandes.edu.co
202320177

Alejandro Salcedo

a.salcedo11@uniandes.edu.co
202321921

Samuel Osorio

sd.osorio2@uniandes.edu.co
202324806

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

La altura del árbol en el que se han almacenado las parejas llave valor de los crímenes crece de forma logarítmica con respecto al número de elementos. Además, tenemos que la cantidad de nodos en el árbol debe oscilar entre $h+1 \leq n \leq 2^{h+1}-1$ e intercambiando los valores tenemos que $29+1 \leq 1177 \leq 2^{29+1}$ resumiendo tenemos, $30 \leq 1177 \leq 1073741824$, por lo que la altura del árbol va concorde a los estándares.

```
Cargando información de crímenes ....
Crímenes cargados: 319073
Altura del arbol: 29
Elementos en el arbol: 1177
Menor Llave: 2015-06-15
Mayor Llave: 2018-09-03
```

¿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?

En caso de hacerlo con tablas de hash, el tiempo sería significativamente mayor debido a que las tablas de hash no son ordenadas, es decir, habría que hacer un recorrido para poder obtener todas las llaves entre ese rango de fechas. Esto causaría que la búsqueda se realizara en un $O(N)$, un tiempo mucho mayor al tiempo utilizando árboles.

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

En la función `getCrimesByRange` en donde se ingresa el rango de las fechas se hace una lista en la que se usa `om.values` el cual obtiene el valor de las llaves en el árbol correspondiente a los rangos de la fecha, después con una iteración en la lista se obtienen los tamaños de los crímenes realizados el mismo día y así consecutivamente hasta culminar la lista.

```
def getCrimesByRange(analyzer, initialDate, finalDate):  
    """  
    Retorna el numero de crímenes en un rango de fechas.  
    """  
    lst = om.values(analyzer['dateIndex'], initialDate, finalDate)  
    totcrimes = 0  
    for lstdate in lt.iterator(lst):  
        totcrimes += lt.size(lstdate['lstcrimes'])  
    return totcrimes
```