

OBSERVACIONES DEL LA PRACTICA

Valeria Caro Ramírez Cod 202111040
Sofia Velasquez Marin Cod 202113334

Preguntas de análisis

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

Respuesta:

N = número de elementos
h = altura

Podemos encontrar el número de elementos con la altura del árbol con la siguiente operación:

$$N = [h + 1, 2^{h+1} - 1]$$

Podemos encontrar la altura del árbol con el número de elementos utilizando la siguiente operación:

$$h = [\text{floor}(\log_2 N), N - 1]$$

Por lo que se puede concluir que la relación es que con la altura del árbol se puede encontrar N y con N se puede encontrar la altura del árbol, por lo que si se tiene cualquiera de las dos variables podemos encontrar la altura o el número de elementos.

- 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?

Respuesta:

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

El tiempo de espera y la complejidad temporal con tablas de hash sería mayor, debido a que las tablas de hash no tienen un orden, por lo que si se quisiera hacer esta misma consulta se debería iterar la lista de todas las llaves de la tabla de hash (keySet) y por cada llave se saca su valor (get(map, key)['value'] ó x = get(map, key) y luego getValue(x)) y luego se verifica que la fecha se encuentre en el rango dado, por lo que la complejidad sería O(#total de llaves).

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

Respuesta:

```
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
```

La operación TAD que se utiliza para retornar la lista con la información encontrada en un rango de fechas, es *values*. La operación *values(map, key_{lo}, key_{hi})* recibe un mapa, key_{lo}(Limite inferior) = InitialDate y key_{hi}(Limite superior) = FinalDate y devuelve una lista con los crímenes que sucedieron entre en el rango de fechas.