

OBSERVACIONES DEL LA PRACTICA

Camilo Molina Plata Cod 202221119
Samuel Rozen Mogollon Cod 202123592
Santiago Macias Cuel Cod
202221028

Preguntas de análisis

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

En un árbol binario equilibrado, la altura (profundidad) del árbol es aproximadamente logarítmica en base 2 con respecto al número de elementos. La altura aumentará lentamente a medida que aumentan los elementos, lo que garantiza un rendimiento eficiente en las operaciones de búsqueda y manipulación. Sin embargo, en un árbol binario desequilibrado, el número de elementos puede acercarse a la altura, lo que resulta en operaciones menos eficientes.

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 sería mayor o menor? ¿Por qué?

Si la información estuviera en tablas de has en lugar de un Binary Search Tree, el tiempo de respuesta para consultas de búsqueda directa sería generalmente menor en una tabla de has debido a que:

1. **Acceso directo en tablas de hash:** Las tablas de hash están diseñadas para proporcionar acceso en tiempo constante promedio $O(1)$, para operaciones de búsqueda, inserción y eliminación. Esto se logra mediante el uso de una función hash que mapea las llaves a índices en la tabla. Si la función hash está bien diseñada y la tabla no está demasiado llena, la mayoría de las búsquedas se pueden realizar con una o dos operaciones.
2. **Búsqueda logarítmica en BST:** En un BST equilibrado, las operaciones de búsqueda, inserción y eliminación tienen un tiempo de complejidad $O(\log n)$, donde n es el número de nodos en el árbol. Esto significa que a medida que el árbol crece, el tiempo que toma realizar estas operaciones crece logarítmicamente.

De todas maneras, hay situaciones en las que un BST podría ser más rápido. Por ejemplo, si necesitamos realizar consultas de rango (tal como encontrar todos los elementos entre dos valores), un BST sería más eficiente que una tabla de hash. Además, hay que asegurarse que la tabla de hash no tenga colisiones y que este bien diseñada, ya que de otra manera su rendimiento podría degradarse.

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

Del siguiente código se ve como del módulo ordered map se implementa la función values para retornar una lista con la información encontrada en un rango de fechas. Esta retorna los valores de

las parejas llave - valor cuya llave esté entre la fecha inicial (initialDate) y la fecha final (finalDate) dentro del mapa ordenado por fechas contenido en el analizador.

```
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
```