

OBSERVACIONES DEL LA PRÁCTICA

Estudiante 1: Nathalia Quiroga, Cod: 202013212

Estudiante 2: David Valderrama, Cod: 201910987

Preguntas de análisis

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

Con una altura de 29 el árbol podría acomodar hasta 1.073.741.823 elementos teniendo en cuenta la fórmula $2^{(h+1)}-1$. Durante la carga, el programa arrojó 1177 elementos acomodados en el árbol: sólo un poco más del 0,0001% de su capacidad máxima, por lo tanto, podemos concluir que es imposible que el árbol esté balanceado, ya que solo se puede garantizar que esté lleno hasta el nivel 9.

- 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?

Se podría decir que el tiempo de respuesta sería menor para realizar una consulta en un árbol binario de búsqueda, porque el BST tiene noción de orden de sus elementos, con esto se facilita trabajar con rangos, pues solo basta con saber los extremos del rango a consultar para obtener la información, mientras que la tabla de hash, al no tener noción de orden y estar orientada a consultas de un único elemento, necesita de algún otro agente que le ayude a encontrar uno por uno los elementos del rango.

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

En el API del TAD mapa ordenado hay una función llamada values que se usó para retornar los crímenes cometidos entre dos fechas dadas por el usuario.

```
def values(map, keylo, keyhi):
    """
    Retorna todas los valores del arbol que se encuentren entre
    [keylo, keyhi]
    Args:
        map: La tabla de simbolos
        keylo: limite inferior
        keylohi: limite superiorr
    Returns:
        Las llaves en el rago especificado
    Raises:
        Exception
    """
    return om.values(map, keylo, keyhi)
```