OBSERVACIONES DEL LA PRACTICA

Juan David Salguero 201923136
David Molina 202125176

Preguntas de análisis

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

La relación es la siguiente:
Elementos: 319073 - Altura del árbol: 29
Ya que se entiende a la altura del árbol la complejidad necesaria para encontrar un elemento en el mismo, se piensa que está relación es impresionante. De una gran cantidad de datos un árbol logra reducir la complejidad a menos del 0.001% del total de iteraciones que se tendría que hacer normalmente en una lista.

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?**

El tiempo de respuesta sería mayor. La solución más eficiente a este problema en un TADMap podría consistir en buscar los crímenes de cada año en el rango dado según un índice de años. No obstante, esto sólo sería posible con un formato que solo tome en cuenta el año de cada crimen. Resultaría imposible con un formato de Año-Mes-Día y la complejidad de traer cada elemento sería mayor que la de iterar un árbol de una altura tan pequeña.

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

Aparte de un ma.get() y un me.getValue() para procesar la información recibida, principalmente se usa la función **om.get(),** importada de *orderedmap*, para iterar el árbol en busca del elemento deseado según un índice que guarda los crímenes de menor a mayor según su fecha.

```python
def getCrimesByRangeCode(analyzer, initialDate, offensecode):
    """
    Para una fecha determinada, retorna el numero de crimenes
    de un tipo especifico.
    """
    crimedate = om.get(analyzer['dateIndex'], initialDate)
    if crimedate['key'] is not None:
        offensemap = me.getValue(crimedate)['offenseIndex']
        numoffenses = m.get(offensemap, offensecode)
        if numoffenses is not None:
            return m.size(me.getValue(numoffenses)['lstoffenses'])
    return 0
```