

# LABORATORIO NO. 7: TABLAS DE SÍMBOLOS ORDENADAS

Jennifer Alejandra Montoya Peñuela 202012739 [j.montoyap@uniandes.edu.co](mailto:j.montoyap@uniandes.edu.co)

Santiago Pardo Bravo 202013024 [s.pardob@uniandes.edu.co](mailto:s.pardob@uniandes.edu.co)

Cristhian David Perdomo Gómez 201316701 [cd.perdomo10@uniandes.edu.co](mailto:cd.perdomo10@uniandes.edu.co)

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

Si el árbol binario está totalmente balanceado podemos establecer la siguiente relación de sus elementos y altura.

$$\sum_{i=1}^h (2^i) = 2^{h+1} - 2$$

Siendo  $h$  la altura del árbol y  $n$  el número de elementos.

Por el contrario, si el árbol se encuentra totalmente desbalanceado la relación entre la altura y sus elementos sería  $h = n$  pues sería una lista.

**Pregunta 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?

Mayor, ya que en este punto nos están solicitando dar la información de todos los accidentes dado un rango de fechas, si se implementara una tabla de hash la búsqueda tendría una complejidad temporal  $O(n)$ , pues prácticamente se requeriría, en el peor de los casos, recorrer todos los accidentes para poder determinar si hacen parte o no del rango de fechas solicitado. Lo anterior debido a que las tablas de hash se implementan sin ordenar. Si la pregunta fuera dirigida a una fecha específica, se podría formar la tabla de hash con las fechas como llaves y en este caso sí podrían obtenerse búsquedas mucho más rápidas.

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

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

DEL ADT ORDEREDMAP

```
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 rango especificado

    Raises:
        Exception
    """
    return om.values(map, keylo, kehi)
```

La siguiente imagen ilustra las estructuras de datos utilizadas.

