**OBSERVACIONES DEL LA PRACTICA**

Sofía López García Cod 201922558

Jonathan Rivera Cod 202022864

# **Preguntas de análisis**

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

Para comentar respecto a la relación que poseen estos dos datos, es necesario explicar que representan cada uno de estos. Por una parte, el número de elementos en el árbol representa el número de nodos que posee este árbol binario. Por otra parte, la altura del árbol representa el número del máximo nivel (donde se encuentra la hoja más baja del árbol) + 1. Ya contando con la definición de los conceptos, se puede deducir de la relación entre número de nodos (1177) y la altura del árbol (29) que el árbol no se encuentra lleno.

Lo anterior se puede entender por que para que un árbol se encuentre lleno, todos los árboles tienen que tener sus dos hijos y por lo tanto todos los niveles tienen que estar llenos, como cada nivel lleno tiene una cantidad de nodos correspondientes a las potencias de 2 (que aumenta cuando se pasa a un nivel mas bajo), el numero total de nodos va a ser la sumatoria de las potencias de 2 hasta llegar a 2\*n, donde n representa la altura del árbol, lo que nos da a entender que la formula para calcular el número máximo de nodos es 2(n+1)-1, y que por lo tanto el máximo de nodos para un árbol que posee una altura de 29 es igual a: 2(29+1)-1 = 2\*30 -1 = 1073741824 – 1= 1073741823. De todo lo anterior se puede concluir que no se trata de un árbol lleno, sin embargo, se necesitaría más información para saber si el árbol se encuentra balanceado.

También, ya que la máxima altura posible del árbol esta representada por n-1, donde n es la cantidad de nodos, se entiende que este árbol binario no se trata de una lista encadenada, que es la estructura que representa el árbol cuando posee su altura máxima, que en este caso seria 1176 (la altura en este caso es 29) Para comentar respecto a la relación que poseen estos dos datos, es necesario explicar que representan cada uno de estos. Por una parte, el número de elementos en el árbol representa el número de nodos que posee este árbol binario. Por otra parte, la altura del árbol representa el número del máximo nivel (donde se encuentra la hoja más baja del árbol) + 1. Ya contando con la definición de los conceptos, se puede deducir de la relación entre número de nodos (1177) y la altura del árbol (29) que el árbol no se encuentra lleno.

Lo anterior se puede entender por que para que un árbol se encuentre lleno, todos los árboles tienen que tener sus dos hijos y por lo tanto todos los niveles tienen que estar llenos, como cada nivel lleno tiene una cantidad de nodos correspondientes a las potencias de 2 (que aumenta cuando se pasa a un nivel mas bajo), el numero total de nodos va a ser la sumatoria de las potencias de 2 hasta llegar a 2\*n, donde n representa la altura del árbol, lo que nos da a entender que la formula para calcular el número máximo de nodos es 2(n+1)-1, y que por lo tanto el máximo de nodos para un árbol que posee una altura de 29 es igual a: 2(29+1)-1 = 2\*30 -1 = 1073741824 – 1= 1073741823. De todo lo anterior se puede concluir que no se trata de un árbol lleno, sin embargo, se necesitaría más información para saber si el árbol se encuentra balanceado.

También, ya que la máxima altura posible del árbol esta representada por n-1, donde n es la cantidad de nodos, se entiende que este árbol binario no se trata de una lista encadenada, que es la estructura que representa el árbol cuando posee su altura máxima, que en este caso seria 1176 (la altura en este caso es 29)

1. ¿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 en una tabla de hash porque la complejidad temporal asociada a la selección de un subconjunto de llaves en un rango (en este caso las fechas) sería muy alta. Esto sucede porque las tablas de hash no tienen orden, entonces no es eficiente utilizarla para implementar tablas de símbolos ordenadas ya que se tiene que recorrer toda la tabla para obtener todos y cada uno de los casos posibles.

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

Se utiliza del TAD map(ordered): values() que es om.values() llamando a orderedmap.py como om

Más específicamente:

En el view:

controller.getCrimesByRange(cont, initialDate, finalDate)

En el controller:

model.getCrimesByRange(analyzer, initialDate.date(),

finalDate.date())

En el model:

def getCrimesByRange(analyzer, initialDate, finalDate):

"""

Retorna el numero de crimenes en un rago de fechas.

"""

lst = om.values(analyzer['dateIndex'], initialDate, finalDate)

totcrimes = 0

for lstdate in lt.iterator(lst):

totcrimes += lt.size(lstdate['lstcrimes'])

return totcrimes

Despues de usar la operación values del TAD ordered maps, se crea un condicional **for-in** donde se usa el TAD lista, ya que el BST tiene como llave una fecha y como valor una pareja de elementos donde uno de los elementos es la lista de todos los crímenes ocurridos en la fecha de la llave.

Para responder a la consulta, se necesita ingresar a los valores (cosa que ya se hizo con la operación om.values() ) y recorrer los elementos de la lista. Para ello se utiliza la operación lt.iterator(), que simplemente retorna un iterador para la lista, y lt.size() que informa el número de elementos de la lista.

**OBSERVACIONES DEL LA PRACTICA No.9**

1. ¿Qué diferencia existe entre las alturas de los dos árboles (BST y RBT)?

La altura del árbol BST es 29 mientras que la del RBT es 13.

1. ¿Por qué pasa esto?

Porque el árbol RBT asegura balance (o un parcial balance si lo que ocaciona el desbalance es un arco rojo, que es aceptado).