

# OBSERVACIONES DE LA PRÁCTICA

1. Samuel Felipe Tovar Palomo, [sf.tovarp1@uniandes.edu.co](mailto:sf.tovarp1@uniandes.edu.co).
2. Juan Esteban Espitia Perez, [je.espitia@uniandes.edu.co](mailto:je.espitia@uniandes.edu.co).
3. Fabio Andres Salazar Ochoa, [fa.salazar@uniandes.edu.co](mailto:fa.salazar@uniandes.edu.co).

## Preguntas de análisis

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

En el árbol binario de búsqueda implementado, la altura depende directamente del orden en que se insertan los elementos. Cuando los datos se insertan de manera aleatoria o balanceada, la altura crece de forma aproximadamente logarítmica con respecto al número de elementos, lo que significa que el árbol mantiene una estructura eficiente y las operaciones básicas (búsqueda, inserción y recorrido) se realizan en un tiempo cercano a  $O(\log n)$ . Sin embargo, si las inserciones se realizan con llaves ordenadas, por ejemplo, fechas de manera ascendente o descendente, el árbol se desbalancea y su altura aumenta casi linealmente con el número de elementos, alcanzando una complejidad de  $O(n)$ .

### 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?*

Si las consultas de rango se realizaran sobre una tabla hash en lugar de un árbol binario de búsqueda, el tiempo sería significativamente mayor. Esto se debe a que las tablas hash no conservan ningún tipo de orden entre las llaves; por tanto, no es posible obtener directamente todos los elementos comprendidos entre dos valores (por ejemplo, entre dos fechas) sin recorrer la totalidad de las entradas. Esto implica una complejidad de  $O(n)$ , ya que cada elemento debe inspeccionarse individualmente para verificar si pertenece al rango. En cambio, en un árbol binario de búsqueda, las llaves están almacenadas de forma ordenada, lo que permite recorrer únicamente el subconjunto de nodos que se encuentran dentro del intervalo, logrando una eficiencia aproximada de  $O(\log n)$ .

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

En la implementación del laboratorio, la operación utilizada para obtener la información de los nodos dentro de un rango de fechas es el método **keys()**, el cual invoca internamente la función recursiva **keys\_range()** definida en el módulo `binary_search_tree.py`. Esta función recorre el árbol comparando cada llave con los límites inferior y superior del rango especificado, e incluye en la lista de salida solo aquellas que se encuentran dentro del intervalo.