

# OBSERVACIONES DE LA PRÁCTICA

Juliana Rodríguez Morales 202421552

María Clara Quijano 202420069

Juan Andrés Lozada Barragán 202510410

## Preguntas de análisis

1. ¿Qué diferencia existe entre las alturas de los dos árboles (BST y RBT)?
  - La altura en un árbol BST puede ser máximo del número de entradas, pero solo es eficiente su implementación si la altura es más baja. La altura ideal para un árbol BST es de  $\lceil \log(n+1) \rceil - 1$ , y en el caso promedio es de  $O(\log n)$ .
  - En el caso del árbol RBT, la altura debe cumplir con  $\log(n+1) - 1 \leq h \leq 2\log(n+1) - 2$ , por lo que se determina que  $h$  es  $O(\log n)$ .
  - El árbol RBT mantiene siempre la misma altura, mientras que el árbol BST puede ser lineal en su peor caso.
2. ¿Percibe alguna diferencia entre la ejecución de los dos árboles (RBT y BST)? ¿Por qué pasa esto?
  - En la ejecución, un árbol BST se puede volver muy lento si el orden de entrada no es favorable, por ejemplo, si es creciente, ya que se va a volver lineal. No tiene nada que asegure que la altura no se vuelva muy alta.
  - El árbol RBT tiene reglas que lo mantienen balanceado, como la rotación y los colores, y asegura que los tiempos por operación sean estables.
3. ¿Existe alguna diferencia de complejidad entre los dos árboles (RBT y BST)? Justifique su respuesta.
  - RBT siempre garantiza una complejidad de  $O(\log n)$  por las funciones que tiene que no permite que la altura sea muy grande, mientras que el árbol BST tiene de peor caso  $O(n)$ . Las operaciones dependen de la altura del árbol y, por lo tanto, es beneficioso mantenerla balanceada.
4. ¿Existe alguna manera de cargar los datos en un árbol RBT de tal forma que su funcionamiento mejore? Si es así, mencione cuál.
  - Se puede ir balanceando el árbol a medida que se va construyendo para no tener que volver a recorrer todos los nodos después de insertar algo y requerir menos tiempo.