

OBSERVACIONES DE LA PRÁCTICA

Simón Peña Alarcón - 202512907- s.penaa2@uniandes.edu.co

Juan Sebastian Chacón Ochoa-202513196-j.chacono@uniandes.edu.co

Manuel Santiago Figueroa-202511697- m.figueroag@uniandes.edu.co

Preguntas de análisis

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

En general, el árbol RBT (Red-Black Tree) tiene una altura menor que el BST normal, porque el RBT se mantiene balanceado automáticamente al insertar los elementos.

En el laboratorio pude ver que la altura del BST crecía mucho más rápido cuando los datos se insertaban en cierto orden, mientras que el RBT se mantenía más estable.

Por ejemplo, con la misma cantidad de datos, el RBT tenía una altura casi logarítmica, mientras que el BST se acercaba a una altura lineal.

2. ¿Percibe alguna diferencia entre la ejecución de los dos árboles (RBT y BST)? ¿Por qué pasa esto?

Sí, el RBT ejecuta más rápido las búsquedas y recorridos cuando la cantidad de datos aumenta.

Esto pasa porque el RBT evita que el árbol se desbalancee, por lo que las operaciones recorren menos niveles en promedio.

En cambio, el BST se puede volver muy alto si los datos llegan en orden (por ejemplo, de menor a mayor), haciendo que cada operación recorra casi todos los nodos.

En el laboratorio se nota que el RBT tarda un poco más al insertar (por los rebalanceos), pero luego las consultas son más rápidas.

3. ¿Existe alguna diferencia de complejidad entre los dos árboles (RBT y BST)? Justifique su respuesta.

La respuesta es si, en un BST normal, la complejidad depende de la forma del árbol:

En el mejor caso balanceado: $O(\log n)$

En el peor caso desbalanceado: $O(n)$

En un RBT, gracias a las rotaciones y cambios de color, el árbol se mantiene casi perfectamente balanceado, garantizando la complejidad $O(\log n)$ en todas las operaciones insertar, buscar y eliminar. Por eso, aunque ambos tienen complejidad promedio similar, el RBT asegura ese rendimiento incluso en los peores casos.

4. ¿Existe alguna manera de cargar los datos en un árbol RBT de tal forma que su funcionamiento mejore? Si es así, mencione cuál.

Si existe. Aunque el RBT se balancea solo, su rendimiento mejora si los datos se insertan de forma más aleatoria o ya pre-ordenados de manera balanceada.

Por ejemplo, si antes de insertarlos se mezclan o se insertan en un orden que no genere muchos rebalanceos seguidos, el tiempo total de inserción es menor.

También ayuda cargar los datos por lotes ordenados y no de forma completamente secuencial de menor a mayor, para que el árbol no tenga que hacer tantas rotaciones iniciales.