**OBSERVACIONES DEL LA PRACTICA**

Los cabezas:

Daniel Rincon Cod 202420898

Nicolas Castano Cod 202420324

Juan David Echeverria Cod 202420422

**Preguntas de análisis**

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

La principal diferencia entre las alturas de un BST y de un RBT es principalmente el balance que existe entre las ramas. Esto es porque un BSt puede ser desbalanceado, lo que significa que pueden haber varios caminos desde las hojas hasta la raíz mucho mas grandes y largo s que otros. En cambio, en un RBT, los nodos y las hojas siempre se añaden o se borran de tal manera que el árbol siempre este balanceado. Su altura siempre debe ser O(log n). Además, el RBT garantiza mayor eficiencia en términos de altura, porque evita el peor escenario de un BSt desbalanceado.

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

Por supuesto que percibimos una gran diferencia entre la ejecucion de los dos arboles. En primer lugar porque el BSt no tiene un mecanismo que garantice que el arbol este balanceado. Esto significa que existen casos donde la estructura es completamente impredecible, incluso podria llegar a tomar una forma de single linked list. En cambio, los RBT, al tener reglas de balanceo, se asegura de que todas las operaciones sean siempre eficientes y consistentes, y que no sea por asi decirlo, una estructura "Adaptativa". Este balance se logra a traves de rotaciones y ajustes en los nodos, operaciones que no existen en los BSTs.

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

Si existe una gran diferencia de complejidad entre los dos arboles. Esto es debido a que principalmente, los BST tienen mejor y peor caso. El mejor caso es que este balanceado, y el peor es que no. EN el caso de que no este balanceado, la complejidad (es decir la altura) pasa a ser O(n). Mientras que en los RBT no existe ni mejor ni peor caso, o mejor dicho son el mismo: que siempre este balanceado. Esto se debe a que los RBt están diseñados para siempre mantenerse balanceados. Basicamente, la diferencia de complejidad entre ambos arboles garantiza un mejor rendimiento en los RBT para todas las operaciones.

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

Sí, existe una manera de cargar los datos en un RBT que puede mejorar su funcionamiento: insertar los datos de manera aleatoria. Si los datos se insertan en un orden aleatorio, es menos probable que se hagan muchas rotaciones a la hora de insertar los nuevos nodos, lo que garantiza que las operaciones de inserción, eliminación y búsqueda mantengan una complejidad de O(log n). Cargar los datos en un orden aleatorio previene los peores casos, donde el árbol podría degenerar en una estructura lineal si los elementos se insertan en orden ascendente o descendente.