

INSERCIÓN

A continuación, tenemos una tabla que compara el tiempo que demora en insertar 100 elementos del género de acción de anime a la estructura:

Estructura	Tiempo(s)
ABB	0,001255
AVL	0,002286
2-3	0,003203

Teóricamente, tenemos que ABB tome tiempo $O(h)$ siendo h la altura de la rama donde corresponde la inserción, AVL toma tiempo $O(\log_2 N)$, con N equivalente al número de nodos en el árbol, siendo este una cota a la altura máxima de la rama debido al balance que logra el árbol, y 2-3 toma tiempo $O(\log N)$, con N equivalente al número de nodos en el árbol, donde la expresión corresponde a una cota de la altura de la rama donde se debe de hacer la inserción.

Debido a la poca cantidad de datos insertados, vemos que se favorece el algoritmo más simple, la inserción de ABB. Esto ocurre ya que las estructuras AVL y 2-3 requieren de una reordenación cada cierta cantidad de inserciones para poder mantener sus propiedades, por lo que se ven en desventaja en escalas pequeñas de datos en comparación con un ABB.

BUSQUEDA

A continuación, tenemos una tabla que compara el tiempo de búsqueda del mismo elemento en diferentes estructuras:

Estructura	Tiempo(s)
ABB	0,000585
AVL	0,000514
2-3	0,000537

Teóricamente, tenemos que ABB tome tiempo $O(h)$ siendo h la altura de la rama donde corresponde se encuentra el elemento, AVL toma tiempo $O(\log_2 N)$, con N equivalente al número de nodos en el árbol, siendo este una cota a la altura máxima de la rama debido al balance que logra el árbol, y 2-3 toma tiempo $O(\log N)$, con N equivalente al número de nodos en el árbol, donde la expresión corresponde a una cota de la altura de la rama donde se debe de hacer la inserción.

En esta operación tenemos que, una vez construido el árbol, las estructuras con una mejor cota a la altura de las ramas de su árbol son más rápidas en las operaciones de búsqueda, ya que el camino que recorren es más corto.

ELIMINACIÓN

A continuación, tenemos una tabla que compara el tiempo de eliminación del mismo elemento en diferentes estructuras:

Estructura	Tiempo(s)
ABB	0,001112
AVL	0,001374
2-3	0,001348

Teóricamente, la complejidad de esta operación de mantiene igual al resto de las operaciones para todas las estructuras.

En este caso, tenemos que la operación de eliminación consta de búsqueda, eliminar el elemento del árbol y luego reordenar el árbol. Para el caso del ABB, la reordenación es $O(1)$, por lo que, si bien su complejidad de búsqueda no es tan buena, le gana al resto de las estructuras al tratar con dataset pequeños. Las otras dos estructuras tienen una complejidad de reordenación de $O(h)$, siendo h la altura de la rama donde estaba el elemento, lo que explica que demoraran mas en esta operación.