Universidad del Bío-Bío
FACULTAD DE CIENCIAS EMPRESARIALES
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION Y TECNOLOGIA DE LA INFORMACIÓN INGENIERÍA CIVIL EN INFORMATICA

TAREA 1: ÁRBOL AVL

Nombre: Diego Loyola

Asignatura: Estructura de datos

Profesor: Alejandro Troncoso

Rotaciones (IZQ Y DER)

Las rotaciones AVL se realizan para mantener la propiedad de equilibrio del árbol. Estas rotaciones tienen un costo constante en términos de tiempo, y su ejecución se basa en las comparaciones de alturas.

Método de Inserción

La complejidad O(log2 n) se cumple cuando el árbol es equilibrado, lo que nos garantiza que en el peor de los casos el número de operaciones de comparación crece logarítmicamente con el número de nodos en el árbol. En este caso cada nivel del árbol el código realiza una sola comparación (**compara = palabra.compareTo(nodo.palabra))** que determina si debe ir hacia la izquierda o hacia su derecha y esta comparación se realiza en tiempo constante.

La recursión sigue el camino apropiado del árbol y la cantidad de niveles a los que se llega antes de encontrar el índice adecuado para la inserción esta limitada por la altura del árbol.

Si nuestro árbol esta equilibrado la altura del árbol está limitada por log2(n), donde "n" es el número de nodos dentro del árbol.

En resumen el código de inserción si cumple con la restricción O(log2 n) si el árbol está equilibrado, si este no se encuentra equilibrado ósea se insertan de manera desordenada esta complejidad pasaría al peor caso que es O(N).

Método de búsqueda

Este método realiza una búsqueda recursiva en el árbol de búsqueda binaria para encontrar un nodo que contenga la palabra deseada. La razón por la cual la función satisface la restricción de $O(\log_2 n)$ es porque, en cada paso de la recursión, se reduce el tamaño del problema a la mitad al elegir el subárbol izquierdo o derecho.

Conclusión:

En conclusión, la implementación sigue las características de un árbol AVL y asegura que el árbol esté equilibrado después de cada inserción, lo que nos garantiza una eficiencia de búsqueda de $O(\log_2 n)$.