Tabla de complejidad para los métodos de la clase Temperaturas_DB

Metodo	Orden de complejidad
guardar_temperatura	O(log_2(n))
devolver_temperatura	O(log_2(n))
max_temp_rango	O(n)
min_temp_rango	O(n)
temp_extremos_rango	O(n)
borrar_temperatura	O(log_2(n))
mostrar_temperaturas	O(n)
mostrar_cantidad_muestras	O(1)

El orden de complejidad para agregar, eliminar o buscar una sola temperatura en el árbol es logarítmico de base 2. Esto se debe a la forma en que se ordena el árbol, donde cada nivel completo tiene el doble de elementos que el nivel anterior. Por lo tanto, la cantidad de operaciones necesarias para llegar a la máxima profundidad del árbol es logarítmica.

En el caso de la función guardar_temperatura, se recorre el árbol en forma de búsqueda para encontrar el lugar adecuado donde guardar la temperatura. En el caso de la función eliminar, se debe recorrer el árbol para encontrar el nodo con el dato a eliminar. Una vez encontrado el nodo, las operaciones de borrado y agregado pueden desequilibrar el árbol, pero se solucionan con las rotaciones, las cuales tienen un orden constante O(1) y no alteran el orden de la acción.

Para las búsquedas en rango, como min_temp_rango, max_temp_rango y temp_extremos_rango, se debe analizar un subárbol completo dentro del árbol. En el peor caso, si el rango abarca todo el conjunto de datos, se deben analizar los n nodos del árbol. Lo mismo ocurre con la función mostrar_temperaturas, que podría considerarse una función que abarca todos los nodos del árbol.

En resumen, el árbol proporciona un orden de complejidad logarítmico para las operaciones individuales y puede requerir analizar todos los nodos en casos específicos de búsquedas en rango.