

Ejercicio 2:

La implementación de las clases `NodoArbol`, `ArbolBinarioBusqueda` y `ArbolAVL` se basa en la bibliografía proporcionada por la cátedra. Sin embargo, en el módulo `ArbolAVL`, realicé modificaciones para habilitar la búsqueda y visualización de temperaturas por rango.

Luego, en la aplicación principal, he creado una instancia llamada `base_datos` de la clase `Temperaturas_DB`. He utilizado esta instancia para agregar, consultar y eliminar medidas de temperatura, además de realizar consultas en rangos de fechas y mostrar los resultados.

Complejidades de las funciones de `Temperaturas_DB`

Función	Complejidad	Descripción
<code>guardar_temperatura</code>	$O(\log n)$	Inserta un nodo en el árbol AVL, con una complejidad promedio de $O(\log n)$ debido al equilibrio del árbol.
<code>devolver_temperatura</code>	$O(\log n)$	Busca un nodo por fecha en el árbol AVL, con una complejidad promedio de $O(\log n)$ gracias al equilibrio del árbol.
<code>max_temp_rango</code>	$O(\log n + k)$	Busca y encuentra la temperatura máxima en un rango de fechas. Requiere buscar el nodo inicial ($O(\log n)$ en promedio) y recorrer 'k' nodos en el rango.
<code>min_temp_rango</code>	$O(\log n + k)$	Busca y encuentra la temperatura mínima en un rango de fechas. Implica buscar el nodo inicial ($O(\log n)$ en promedio) y recorrer 'k' nodos en el rango.
<code>temp_extremos_rango</code>	$O(\log n + k)$	Busca y encuentra tanto la temperatura mínima como la máxima en un rango de fechas. Requiere buscar el nodo inicial ($O(\log n)$ en promedio) y recorrer 'k' nodos en el rango.

<code>borrar_temperatura</code>	$O(\log n)$	Elimina una temperatura por fecha, implicando una búsqueda del nodo ($O(\log n)$ en promedio) y la operación de eliminación en el árbol AVL equilibrado.
<code>devolver_temperaturas</code>	$O(\log n + k)$	Busca y recupera todas las temperaturas en un rango de fechas. Requiere buscar el nodo inicial ($O(\log n)$ en promedio) y recorrer 'k' nodos en el rango.
<code>cantidad_muestras</code>	$O(1)$	Devuelve la cantidad de muestras en la base de datos y no implica operaciones en el árbol. La complejidad es constante, $O(1)$.

Un árbol AVL es una estructura de datos de búsqueda que mantiene su equilibrio para garantizar un rendimiento eficiente en las operaciones de búsqueda, inserción y eliminación. Sin embargo, en el peor de los casos, cuando el rango abarca todo el conjunto de datos, las operaciones que buscan o recorren un rango pueden degradarse a una eficiencia de $O(n)$. Esta situación es más probable en operaciones como `max_temp_rango`, `min_temp_rango`, `temp_extremos_rango` y `devolver_temperaturas`.