**Informe sobre la Implementación de un Árbol AVL para el Almacenamiento y Manipulación de Datos de Temperatura**

**Introducción**

El presente informe describe la implementación de una estructura de datos de árbol AVL para el almacenamiento y manipulación de datos de temperatura junto con sus fechas correspondientes. Un árbol AVL es una estructura de datos de árbol binario balanceado que garantiza que la diferencia de alturas entre los subárboles izquierdo y derecho de cualquier nodo sea como máximo 1. Esto permite realizar operaciones de búsqueda y actualización de manera eficiente.

**Descripción del Código**

**El código implementa dos clases principales:**

NodoAVL: Esta clase representa un nodo en el árbol AVL. Cada nodo contiene la fecha, la temperatura, punteros a los nodos hijos izquierdo y derecho, y la altura del nodo en el árbol.

Temperaturas\_DB: La clase Temperaturas\_DB utiliza un árbol AVL para almacenar y gestionar las temperaturas.

**Pruebas y Resultados**

El código incluye pruebas de los métodos implementados para demostrar su funcionamiento. A continuación, se presentan los resultados de estas pruebas:

Inserción de Datos: Se insertan cuatro temperaturas junto con sus fechas en la base de datos.

Búsqueda de Temperatura por Fecha: Se consulta una temperatura para una fecha específica, y se obtiene la temperatura correctamente.

Consulta de Temperaturas en un Rango de Fechas: Se consultan y se muestran las temperaturas dentro de un rango de fechas específico.

Búsqueda de Temperatura Máxima y Mínima en un Rango de Fechas: Se encuentra tanto la temperatura máxima como la mínima en un rango de fechas y se muestran los resultados.

Búsqueda de Temperaturas Extremas en un Rango de Fechas: Se encuentran tanto la temperatura máxima como la mínima en un rango de fechas y se muestran los resultados.

Eliminación de Datos: Se elimina una temperatura asociada a una fecha específica y se confirma que se ha eliminado.

Cantidad de Muestras Almacenadas: Se consulta la cantidad de muestras almacenadas en la base de datos, y se muestra el número.

**Conclusión**

La implementación de un árbol AVL para el almacenamiento y manipulación de datos de temperatura resulta ser una solución eficiente y organizada. Este enfoque permite realizar búsquedas y consultas en un rango de fechas con rapidez y precisión. Además, la estructura de datos garantiza que el árbol se mantenga balanceado, lo que contribuye a un rendimiento óptimo en términos de tiempo de ejecución.

El código y las pruebas realizadas demuestran que la implementación funciona según lo previsto y es apta para el propósito de gestionar datos de temperatura. Los métodos de búsqueda, eliminación y consulta funcionan de manera coherente y eficiente.

|  |  |  |
| --- | --- | --- |
| **Método** | **Orden de Complejidad** | **Explicación** |
| altura | O(1) | La altura de un nodo ya está almacenada en el nodo mismo, por lo que obtenerla es una operación constante |
| actualizar\_altura | O(1) | La actualización de la altura de un nodo también es una operación constante. |
| balance | O(1) | Calcular el balance de un nodo implica restar las alturas de sus subárboles, una operación constante. |
| rotar\_izquierda | O(1) | Las rotaciones a la izquierdas son operaciones constantes, no dependen del tamaño del árbol. |
| rotar\_derecha | O(1) | Las rotaciones a la derecha son operaciones constantes, no dependen del tamaño del árbol. |
| insertar | O(log n) | La inserción en un árbol AVL tiene una complejidad de O(log n), donde n es el número de nodos en el árbol, debido a las rotaciones que pueden ser necesarias para mantener el equilibrio. |
| guardar\_temperatura | O(log n) | tiene una complejidad de O(log n), ya que llama a insertar, Estas operaciones implican navegar a través del árbol desde la raíz hasta el nodo deseado. |
| devolver\_temperatura | O(log n) | La búsqueda en un árbol AVL tiene una complejidad de O(log N) debido a que se realiza de manera similar a la inserción, descendiendo por el árbol de forma balanceada. |
| max\_temp\_rango | O(n \* log n) | Donde n es el número de temperaturas en el rango. La búsqueda y comparación para encontrar la temperatura máxima ocurren para cada temperatura en el rango. |
| min\_temp\_rango | O(n \* log n) | Donde n es el número de temperaturas en el rango. La búsqueda y comparación para encontrar la temperatura mínima ocurren para cada temperatura en el rango. |
| temp\_extremos\_rango | O(n \* log n) | Donde n es el número de temperaturas en el rango. La búsqueda y comparación para encontrar las temperaturas y ordenarlas de la mínima hasta la máxima, ocurren para cada temperatura en el rango. |
| borrar\_temperatura | O(log n) | La eliminación en un árbol AVL tiene una complejidad de O(log N), similar a la inserción. |
| cantidad\_muestras | O(1) | Obtener la cantidad de muestras es una operación constante, ya que el contador de muestras se mantiene actualizado en cada operación de inserción y eliminación |