Universidad Nacional de Entre Ríos

Facultad de Ingeniería

Algoritmos y Estructuras de Datos

Informe General del Trabajo Práctico N°2

Aplicaciones de conceptos teóricos sobre estructuras jerárquicas, grafos y sus algoritmos asociados

Rodriguez Esteban

Tarabini Melina

2025

Problema 2: “Temperaturas_DB”
Para la resolución de esta consigna se desarrollaron dos módulos: uno correspondiente a la implementación del árbol AVL (AVL.py) y otro que representa la base de datos de temperaturas (Temperaturas_DB.py).

La clase Temperaturas_DB permite almacenar temperaturas asociadas a fechas, utilizando como estructura subyacente un árbol binario AVL, una estructura balanceada de búsqueda binaria que mantiene ordenados los datos cronológicamente (por fecha) y garantiza complejidad logarítmica en las operaciones básicas.

Se implementaron los siguientes métodos:

- **guardar_temperatura(fecha, temperatura):** inserta la temperatura correspondiente a una fecha dada.
- **devolver_temperatura(fecha):** devuelve la temperatura almacenada para la fecha ingresada.
- **max_temp_rango(fecha1, fecha2):** recorre todas las fechas del rango y obtiene la mayor temperatura.
- **min_temp_rango(fecha1, fecha2):** recorre todas las fechas del rango y obtiene la menor temperatura.
- **temp_extremos_rango(fecha1, fecha2):** devuelve una tupla con las temperaturas mínima y máxima encontradas en el rango especificado.
- **borrar_temperatura(fecha):** elimina una entrada del árbol AVL correspondiente a la fecha indicada.
- **devolver_temperaturas(fecha1, fecha2):** devuelve una lista con las temperaturas registradas dentro del rango de fechas dado.

La estructura del árbol AVL, contenida en el archivo AVL.py, asegura que todas las operaciones de inserción, búsqueda y eliminación se realicen en tiempo O(log n), manteniendo balanceado el árbol después de cada modificación.

Método	Descripción	Complejidad Temporal
guardar_temperatura	Inserta una temperatura asociada a una fecha en el árbol AVL	O(logn)
devolver_temperatura	Devuelve la temperatura correspondiente a una fecha dada	O(logn)
max_temp_rango	Recorre el rango de fechas y devuelve la temperatura máxima	$O(k \cdot logn)$
min_temp_rango	Recorrerá el rango de fechas para obtener la mínima	$O(k \cdot logn)$
borrar_temperatura	Elimina la temperatura correspondiente a una fecha	O(logn)
cantidad_muestras	Devuelve la cantidad de muestras de la base de datos	O(1)
devolver_temperaturas	Devuelve un listado ordenado de temperaturas entre dos fechas	$O(k \cdot logn)$
temp_extremos_rango	Devuelve la temperatura mínima y máxima entre dos fechas	$O(k \cdot logn)$

Tabla 2.

Donde n es la cantidad total de temperaturas almacenadas y k es la cantidad de días en el rango especificado.

Para probar el funcionamiento de los métodos de Temperaturas_DB, se implementó un archivo main,py, donde se cargan distintas temperaturas en fechas espaciadas y se realiza una consulta utilizando max_temp_rango. La salida del programa permite verificar que el valor retornado sea correcto.

En el módulo AVL.py hay una prueba para ver las rotaciones que se producen cuando se agregan nodos al árbol .Si bien no es un ensayo exhaustivo de todas las posibilidades, nos permitió verificar su funcionamiento.