

Trabajo Práctico N°2:

Informe Actividad n°2:

Mediante el archivo “muestras”, que es un archivo .txt, que por cada renglón tiene 2 datos, una fecha y la temperatura de ese mismo día, ambos datos separados por “;”. El objetivo de este ejercicio es utilizar diversas funciones para poder obtener datos en base al archivo proporcionado por la cátedra, dichas funciones son:

- **guardar_temperatura(temperatura, fecha):** guarda la medida de temperatura asociada a la fecha.
- **devolver_temperatura(fecha):** devuelve la medida de temperatura en la fecha determinada.
- **max_temp_rango(fecha1, fecha2):** devuelve la temperatura máxima entre los rangos fecha1 y fecha2 inclusive (fecha1 < fecha2). Esto no implica que los intervalos del rango deban ser fechas incluidas previamente en el árbol.
- **min_temp_rango(fecha1, fecha2):** devuelve la temperatura mínima entre los rangos fecha1 y fecha2 inclusive (fecha1 < fecha2). Esto no implica que los intervalos del rango deban ser fechas incluidas previamente en el árbol.
- **temp_extremos_rango(fecha1, fecha2):** devuelve la temperatura mínima y máxima entre los rangos fecha1 y fecha2 inclusive (fecha1 < fecha2).
- **borrar_temperatura(fecha):** recibe una fecha y elimina del árbol la medición correspondiente a esa fecha.
- **devolver_temperaturas(fecha1, fecha2):** devuelve un listado de las mediciones de temperatura en el rango recibido por parámetro con el formato “dd/mm/aaaa: temperatura °C”, ordenado por fechas.
- **cantidad_muestras():** devuelve la cantidad de muestras de la BD.

Para poder hacer todo de una forma más organizada, decidimos organizar el algoritmo en 2 principales módulos:

- AVL (Que maneja la lógica de un árbol AVL)
- Temperaturas_DB (Usada para poder trabajar con las temperaturas proporcionadas)

Finalmente, pudimos realizar cada función como corresponde, y analizamos el orden de complejidad de cada una:

Tabla de ordenes de complejidad:

<code>guardar_temperatura(temperatura, fecha)</code>	$O(\log n)$.
<code>devolver_temperatura(fecha)</code>	$O(\log n)$
<code>max_temp_rango(fecha1, fecha2)</code>	$O(n)$
<code>min_temp_rango(fecha1, fecha2)</code>	$O(n)$
<code>temp_extremos_rango(fecha1, fecha2)</code>	$O(n)$.
<code>borrar_temperatura(fecha)</code>	$O(\log n)$.
<code>devolver_temperaturas(fecha1, fecha2)</code>	$O(n)$.
<code>cantidad_muestras()</code>	$O(n)$

Donde **n** es la cantidad de muestras, o nodos del árbol, y **k** es es la cantidad de elementos devueltos del rango.

Aclaración: El árbol tiene una eficiencia logarítmica en búsquedas individuales, pero todos los métodos que trabajan con rangos lo hacen con un orden de complejidad $O(n)$ al recorrer la lista entera.

Finalmente utilizamos el bloque `if __name__ == "__main__"` para poder probar el correcto funcionamiento de los métodos de Temperatura DB.