

Práctica 7: Implementación de un árbol binario de búsqueda balanceado (AVL)

1. Objetivo

En esta práctica se trabaja la estructura de datos AVL y sus algoritmos, tanto la implementación en lenguaje C++ como el estudio de su rendimiento.

2. Entrega

Esta práctica se realizará en dos sesiones de laboratorio en las siguientes fechas:

- Sesión tutorada: 14, 15 y 16 de mayo de 2019.
- Sesión de entrega: 21, 22 y 23 de mayo de 2019.

Durante las sesiones de laboratorio se podrán proponer modificaciones y mejoras en el enunciado de la práctica.

3. Enunciado

Desarrollar un tipo de dato genérico en lenguaje C++ que implementa el árbol binario de búsqueda balanceado (AVL) [1][2] y realizar un programa en C++ que permita observar su funcionamiento.

Realizar un estudio empírico del rendimiento de un AVL cuando se varía el número de nodos del árbol.

El estudio del rendimiento requiere implementar un programa en C++ que cuente el número de operaciones de comparación de clave que se realizan durante una operación de búsqueda o inserción en el AVL (según el procedimiento descrito en la práctica 4 para la Tabla Hash).

El programa utilizará un contador de comparaciones, que se inicializa a cero antes de la ejecución de cada operación y se incrementa con cada comparación de claves. Al finalizar la operación, el valor contenido en el contador se utiliza para actualizar una estadística que registra los valores mínimo, máximo y media del número de comparaciones obtenidos. Para que estos valores estadísticos sean significativos el experimento debe repetirse un número suficiente de veces. En la sección de notas de implementación se indica el formato de visualización de los datos obtenidos en la ejecución del programa.

El estudio del rendimiento del AVL consiste en realizar distintas ejecuciones del experimento incrementando el tamaño del árbol y anotar los valores mínimo, media y máximo del número de comparaciones obtenidos en cada ejecución.

4. Notas de implementación

Desarrollar en lenguaje C++ la plantilla de clases para el nodo AVL (`nodoAVL<Clase>`) y para el tipo abstracto de dato árbol binario balanceado (`AVL<Clase>`), que implemente las siguientes operaciones:

- Buscar (`Clave x`)
- Insertar (`Clave x`)
- Eliminar (`Clave x`)

Para reutilizar las operaciones comunes a los distintos tipos de árboles, se recomienda derivar las clases `nodoAVL<Clase>` y `AVL<Clase>` de las clases implementadas en la práctica 6 (`nodoBB<Clase>` y `ABB<Clase>`).

Para probar el funcionamiento del AVL y realizar el estudio de su rendimiento se utilizarán valores de clave del tipo `DNI` (clase definida en el enunciado de la práctica 4).

Se deben realizar dos programas para ejecutar el código del ABB implementado:

1. **Programa Modo Demostración:** para verificar el funcionamiento del árbol. Se trabajará con árboles de tamaño limitado para permitir una correcta visualización. El programa realizará la siguiente secuencia de pasos:

1. Generar un AVL vacío.
2. Presentar un menú con las siguientes opciones:

```
[0] Salir  
[1] Insertar Clave  
[2] Eliminar Clave
```

3. Para cada inserción o eliminación, solicitar el valor de clave y realizar la operación en el AVL.
4. Tras cada operación, mostrar el árbol resultante mediante un recorrido por niveles. En cada nivel se muestran los nodos de izquierda a derecha. El subárbol vacío se visualiza con `[.]`. Se utilizará el mismo formato de visualización que en la práctica 6.

2. **Programa Modo Estadística:** El programa realizará la siguiente secuencia de pasos:

1. Solicitar los parámetros para el experimento:
 - a. `N`: Tamaño del árbol.
 - b. Número de pruebas, `nPruebas`: Número de repeticiones de la operación, inserción o búsqueda, que se realiza en el experimento.
2. Crear un banco de prueba con $2 * N$ valores de tipo `DNI` generados de forma aleatoria. El banco de pruebas se guarda en un vector.
3. Generar un AVL e insertar los `N` primeros valores del banco de prueba.
4. El experimento para estudiar el comportamiento de la operación de búsqueda consiste en:
 - a. Inicializar a cero los contadores de comparaciones de claves: valores mínimo, acumulado y máximo.
 - b. Realizar la búsqueda de las `nPruebas` claves extraídas de forma aleatoria de las primeras `N` claves del banco de prueba, o sea, de las claves ya insertadas en el AVL. Para cada búsqueda se cuenta el número de

comparaciones realizadas, y se actualizan los valores mínimo, máximo y acumulado.

- c. Al finalizar el experimento se presentan los valores mínimo, máximo y medio del número de comparaciones de claves contabilizados.
5. El experimento para estudiar el comportamiento de la operación de inserción se basa en contar el número de comparaciones realizado para buscar claves que no se encuentran en el árbol. Consiste en:
- a. Inicializar a cero los contadores de comparaciones de claves. Valores mínimo, acumulado y máximo.
 - b. Realizar la búsqueda de las nPruebas claves extraídas de forma aleatoria de las últimas N claves del banco de prueba, o sea, de las claves que no están insertadas en el AVL. Para cada búsqueda se cuenta el número de comparaciones realizadas, y se actualizan los valores mínimo, máximo y acumulado.
 - c. Al finalizar el experimento se presentan los valores mínimo, máximo y medio del número de comparaciones de claves contabilizados.

A continuación se muestra el formato de salida con los resultados de la ejecución:

Número de Comparaciones					
	N	Pruebas	Mínimo	Medio	Máximo
Búsqueda	xxx	xxx	xxxx	xxxx	xxxx
Inserción	xxx	xxx	xxxx	xxxx	xxxx

5. Referencias

[1] Apuntes de clase.

[2] https://es.wikipedia.org/wiki/Árbol_binario_de_búsqueda