

Grado en Ingeniería Informática
Algoritmos y Estructuras de Datos Avanzadas

Practica 6: Implementación de un árbol binario de búsqueda balanceado (AVL)

1. Objetivos

Desarrollar un tipo de dato genérico en lenguaje C++ que implemente un árbol binario de búsqueda balanceado (AVL).

Realizar un análisis de rendimiento del AVL al variar el número de nodos del árbol.

2. Entrega

Esta práctica no incluye sesión tutorada. Se podrá realizar la entrega durante las dos últimas semanas del segundo cuatrimestre:

Sesión de entrega: 9 al 12 de mayo de 2017

16 al 19 de mayo de 2017

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

3. Enunciado

Desarrollar en lenguaje C++ la plantilla de clases para el nodo AVL (`nodoAVL<Clase>`) y para el tipo abstracto de dato AVL (`AVL<Clase>`), que implemente las operaciones de búsqueda, inserción y eliminación de una clave X en el AVL.

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 5 (`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 Matricula, que representa las matrículas de los vehículos a motor (clase definida en el enunciado de la práctica 3).

Se deben realizar dos programas para ejecutar el código implementado:

1. **Modo demostración:** El programa crea un AVL vacío, y presenta un menú con las siguientes opciones:

[0] Salir

[1] Insertar clave

[2] Eliminar clave

Grado en Ingeniería Informática
Algoritmos y Estructuras de Datos Avanzadas

Tras cada operación se mostrará el árbol resultante mediante un recorrido por niveles. La visualización del árbol se realizará con el mismo formato que en la práctica 5: en cada nivel, se muestran los nodos de izquierda a derecha y el subárbol vacío se visualiza con [.].

En este programa se trabajará con árboles de tamaño limitado para permitir una correcta visualización.

2. **Modo estadística:** Para realizar el análisis del rendimiento de un AVL es necesario contar el número de operaciones de comparación de claves que se llevan a cabo al ejecutar una búsqueda. Esto implica utilizar un contador de comparaciones, que se inicia a cero antes de una búsqueda en el árbol y se incrementa con cada ejecución del operador de comparación de claves. Al finalizar cada búsqueda en el AVL el valor obtenido en el contador de comparaciones se utiliza para actualizar una estadística que registra los valores mínimo, máximo y medio del número de comparaciones observados.

El programa realizará la siguiente secuencia de pasos:

1. Solicita los parámetros para crear un AVL:
 - a) N: Tamaño del árbol (número de nodos 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 $2 \cdot N$ valores de tipo Matricula generados de forma aleatoria. El banco de pruebas se almacena 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.

Grado en Ingeniería Informática
Algoritmos y Estructuras de Datos Avanzadas

- 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:

	<u>Numero de Comparaciones</u>				
	<u>N</u>	<u>P</u>	<u>MínimoMedio</u>		<u>Máximo</u>
Búsqueda	xxx	xxx	xxxx	xxxx	xxxx
Insertión	xxx	xxx	xxxx	xxxx	xxxx

De forma opcional se puede utilizar el programa desarrollado para realizar un estudio sobre el comportamiento de un AVL cuando se incrementa el tamaño del árbol.

4. Referencias

[1] Apuntes de clase

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