**Desarrollo Del Laboratorio: Implementación De Un Árbol Binario De Búsqueda**

Camilo Morales

Fundación Universitaria Internacional De La Rioja

ingeniería informática

Tunja, Colombia

20 De abril Del 2023

Laboratorio: Implementación de un árbol binario de búsqueda

**Preparación del laboratorio**

Como herramienta básica se necesita un **IDE** para poder desarrollar el código pedido. Se recomienda, como herramienta propuesta para otros desarrollos en la asignatura, la utilización de **Eclipse**.

**Descripción del laboratorio**

La actividad consiste en la implementación básica de un árbol binario de búsqueda (ABB) sobre cualquier estructura de datos (como soporte), que el alumno considere oportuno. Las operaciones que hay que realizar son:

* Recorrido en preorden, inorden y posorden (sin imprimir). Realizado.
* Impresión de los elementos del árbol. Realizado con los métodos anteriores del 1 punto (preorden, inorden y posorden).
* Inserción de elementos. Realizado con nombre insertar.
* Borrado de elementos. Realizado con nombre eliminar.

Además, se debe incluir una **clase de prueba** que permita comprobar el correcto funcionamiento de todas las operaciones.

**Entrega del laboratorio**

El estudiante deberá entregar:

* El código fuente (proyecto Eclipse o equivalente).

RTA: Se anexa código de proyecto desarrollado en NetBeans.

* Una documentación técnica de la aplicación.

RTA: luego de tener un análisis previo de la teoría sobre arboles binarios y su desarrollo a nivel de código se inicia con su implementación creando la clase nodo y definiendo los constructores y punteros de cada hijo de árbol que después serán implementados en la nueva clase llamada ArbolTAD en la cual se empiezan a crear los métodos para interactuar con los punteros de la clase nodo, en esta clase empezamos con cada método requerido y su funcionalidad siempre teniendo en cuenta como parámetro inicial divide y vencerás ya que tenemos varios métodos a desarrollar entonces esta metodología es perfecta para implementar en estos casos, entonces luego de haber creado cada método para ejecutar y saber que está en completa funcionalidad creamos una clase main de nombre ArbolBinario\_1 que contendrá el llamado a cada método y de forma que se crea un menú.

Siempre teniendo presente que un árbol binario siendo una estructura de datos que tiene por cada nodo máximo dos hijos con las siguientes propiedades:

1. El subárbol izquierdo de un nodo contiene solo valores menores que el nodo raíz.
2. El subárbol derecho de un nodo contiene solo valores mayores que el nodo raíz.

En total en el laboratorio se piden 6 métodos, pero para el método borrado de elementos se debieron crear e implementar otros 2 métodos que son sucesor y predecesor.

**Método Inserción de elementos.**

Este método de tipo público es para insertar un nuevo nodo al árbol, pero para esto primero debe validar si la raíz del árbol está vacía, luego de que esta condición se cumpla define el nuevo dato como la raíz del árbol y a partir de esta raíz empiezan a funcionar los ciclos que tiene el método pero ya para agregar los nodos hijos del árbol basándose en las características que se tiene en cuanto a los hijos para el árbol binario esto quiere decir que la inserción siempre se realiza de forma recursiva en el árbol para tener un mejor orden respecto a la raíz que se tiene.

**Método Impresión de los elementos del árbol.**

En este método se recorre el árbol y se imprimen sus elementos para esto usamos los 3 métodos del 1 punto los cuales nos presentan 3 diferentes formas de recorrer el árbol y luego de esto ser impreso.

**Método borrado de elementos.**

Este método fue creado como su nombre lo indicia para eliminar un nodo del árbol pero se tiene diferentes casos de eliminación basándonos en que si un árbol tiene hijos o por el contrario no tiene hijos por ende no tiene un predecesor ni tampoco un sucesor, entonces tiene esto claro y mediante el uso de la recursión el método empieza a buscar el dato por ende nos pide el valor del nodo a eliminar luego de esto valida si el árbol tiene hijos en caso de tenerlos entonces busca al sucesor pero a su vez también debe buscar al predecesor del sucesor del nodo a eliminar, para así cambiar el valor y que no se pierda le estructura del árbol binario entonces se elimina el dato y seguido de esto se reemplaza con el sucesor de este y predecesor del sucesor, pero en caso de que el árbol no tenga hijos entonces se declara como nulo ya que carece de sucesor y predecesor todo esto lo hace usando ciclos if, while y llamando los dos métodos creados que son sucesor y predecesor.

**Método Preorden**

Este método lo primero que hace recorrer la raíz y luego a sus hijos en orden desde el más a la izquierda y luego los de la derecha, entonces verifica la raíz y luego recorre en caso de que el árbol no tenga hijos solo se imprime la raíz.

**Método Inorden**

En este método lo que se hace para recorrer el árbol es que se empieza desde el nodo que encontramos mas a la izquierda, luego se va a la raíz y finaliza con el lado derecho, entonces primero verifica si el nodo existe para recorrerlo de la manera ya antes mencionada, en caso de que el árbol no tenga hijos solo se imprime la raíz.

**Método Postorden**

El recorrido de este método consiste en visitar primero los hijos y luego la raíz esto quiere decir que primero empieza por el hijo mas a la izquierda, luego se va a los hijos de la derecha y por último visita la raíz, en caso de que el árbol no tenga hijos solo se imprime la raíz.

En conclusión, estos métodos de recorrido son muy parecidos y fáciles de implementar lo único que cambia un poco es la forma de desarrollo y el orden.

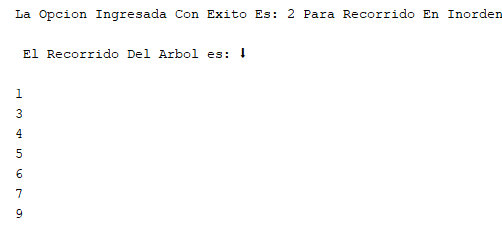
**Clase prueba**

Anexo pantallazos de la clase prueba del código en la página siguiente…

Tabla

Descripción generada automáticamente con confianza mediaIngresa nodo.

Mostrar recorrido del árbol en Inorden.

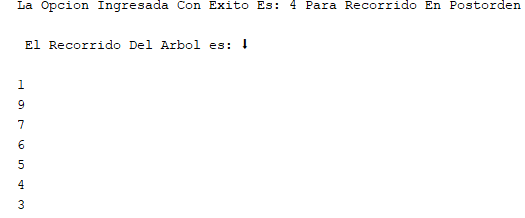


Mostrar recorrido del árbol en Preorden.

Texto

Descripción generada automáticamente

Mostrar recorrido del árbol en Postorden.



Eliminación del número 9

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**Conclusión**

Sin dejar de lado los errores que se tuvieron a la hora de ir ejecutando cada método y el tiempo que llevo solucionarlos para así llegar a tener un muy buen trabajo en cuanto a la parte de código considero que ha sido un trabajo muy interesante de mucho análisis e implementación de herramientas para lograr un muy buen desarrollo de este.

* Un archivo ejecutable (formato JAR), que cumpla con las especificaciones del documento de descripción de la actividad.

RTA: anexo java doc y .JAR