

ESTRUCTURA DE DATOS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y LA DECISIÓN
MARIA C. TORRES
2023-2S

PRACTIA 3: ARBOLES BINARIOS DE BUSQUEDA

Objetivo: implementar y analizar las operaciones de ingresar, eliminar y buscar en los arboles binarios de busqueda.

Recursos: laptop o computadora personal

Lenguaje de programación: Python o JAVA

Instrucciones: realice una implementación en uno de los lenguajes de programa de alto nivel que cumpla con las siguientes especificaciones técnicas. Realice las pruebas necesarias que permitan validar la implementación. Esta práctica se puede desarrollar en grupos de hasta 3 integrantes, todos los integrantes deben participar de la sustentación durante la entrega. La implementación no debe enviarse por correo electrónico ni a través de la plataforma virtual. Cada estudiante o grupo debe presentar en una sustentación de no más de 10 minutos las diferentes funcionalidades.

Fecha de entrega: la implementación debe ser presentada a más tardar el 16 de noviembre, durante las horas de clase, en horas de oficina, o durante la sesión virtual programada el jueves 16 de noviembre.

Requerimientos:

Parte 1: Implementación de Árbol Binario de Busqueda

Implemente en el lenguaje de programación seleccionado las clases necesarias para el funcionamiento de un árbol binario de búsqueda. Los nodos del ABB deben permitir almacenar un objeto genérico, asociado con una clave de tipo entero. El árbol debe mantener la estructura de un ABB; es decir:

- El subárbol izquierdo debe ser un árbol binario de búsqueda
- El subárbol derecho debe ser un árbol binario de búsqueda
- Las claves en el subárbol izquierdo deben ser menor igual a la clave de la raíz
- Las claves en el subárbol derecho debe ser mayor o igual a la clave de la raíz

La implementación debe permitir realizar las siguientes operaciones:

- Insertar un nuevo objeto: dado un objeto con su respectiva clave, el sistema debe ingresar el objeto en la posición correcta del árbol para mantener las propiedades de árbol binario de búsqueda
- Eliminar un objeto dada la clave: dada una clave entera, el sistema debe buscar el objeto en el árbol, y si existe debe eliminar el nodo, manteniendo las propiedades de árbol binario de búsqueda

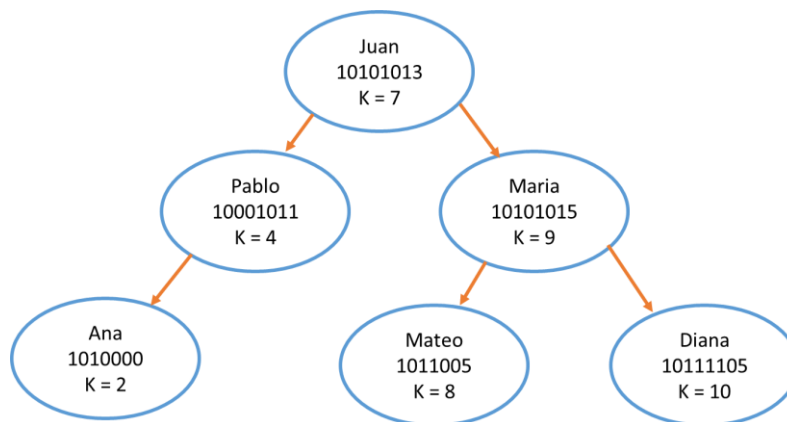
- Buscar un objeto dada la clave: dada una clave entera, el sistema buscar el objeto respectivo y retornar el nodo completo.
- Valor máximo: cuando se llame este método, el sistema debe retornar el objeto con la clave mas grande en el árbol
- Valor mínimo: cuando se llame este método, el sistema debe retornar el objeto con la clave mas pequeña en el árbol.
- Mostrar árbol: este método debe permitir visualizar el árbol en la pantalla. Presente una forma ingeniosa de mostrar los datos del árbol y su estructura.
- Recorrido inorder: este método debe mostrar en pantalla las claves de los nodos cuando se realiza el recorrido inorder.

Parte 2: Prueba de la implementación de ABB

Para probar su implementación de árbol binario de búsqueda, construya un objeto que almacene el nombre y numero de identificación. Asumiremos la clave como el numero entero resultante de sumar los dígitos del numero de identificación. Realice la prueba de los diferentes métodos, ingresando usuario a su ABB.

Por ejemplo, para los siguientes usuarios el árbol binario de búsqueda resultante debe ser:

- Juan 10101013 k=7
- Pablo 10001011 k = 4
- Maria 10101015 k = 9
- Ana 1010000 k=2
- Diana 10111105 k = 10
- Mateo 10110005 k = 8



Notas importantes:

1. El calculo de la clave debe realizarse por un método del objeto
2. En la sustentación se proporcionaran los nombre y números de identificación de otros usuarios, la este es solo un ejemplo

3. Realice la prueba de cada una de las funcionalidades solicitadas en la parte 1. Puede usar su método inorden o de mostrar el árbol para comprobar que las funciones de agregar y eliminar es correcta.