

Instituto Tecnológico de Costa Rica

Curso:

Algoritmos y Estructuras de Datos I

Estudiantes:

José Alejandro Chavarría Madriz

Ignacio Grané Rojas

Natalia González Bermúdez

Profesor:

Diego Noguera

II Semestre 2019

Árbol Binario de Búsqueda

Definición:

Es una estructura de datos no lineal utilizada en la programación que permite almacenar y encontrar datos combinando las ventajas de un arreglo y una lista enlazada. Respectivamente búsquedas rápidas y ser dinámica, es decir, no hay límite para el crecimiento de la estructura y además, permite insertar y eliminar fácilmente sus datos.

Un árbol binario de búsqueda debe cumplir con las siguientes condiciones:

- Cada nodo tienen un máximo de 2 hijos, uno izquierdo y uno derecho.
- El hijo izquierdo siempre es menor que el nodo padre y el derecho siempre es mayor.

Eliminar en Árboles Binarios de Búsqueda

Explicación

Se trata de la acción más compleja en este tipo de árboles. El objetivo de la función consiste en eliminar un nodo específico del árbol binario de búsqueda de tal forma que se mantengan las características, antes mencionadas, de esta estructura.

Cuando se ejecuta este método primero se busca el nodo a eliminar en el árbol. Una vez encontrado el nodo a eliminar existen tres posibles situaciones para el nodo que se desea eliminar:

Casos en la eliminación de un nodo

El nodo es un nodo hoja:

Este es el caso más sencillo ya que solamente se elimina el nodo. El padre del nodo a eliminar va a apuntar a nulo dependiendo si es su hijo izquierdo o derecho el que se quiere eliminar.

Ver ejemplo en figura 1 de anexos.

El nodo posee un nodo hijo:

En este caso tenemos un nodo que presenta un único hijo (nodo nieto), ya sea izquierdo o derecho. Para eliminar este nodo tan solo es necesario un cambio en el apuntador del nodo padre de nuestro nodo a eliminar (nodo abuelo) . Se toma la cambia la referencia que posee el nodo abuelo, la cual pasa de apuntar al nodo a eliminar a el nodo nieto, así efectivamente la referencia al nodo se pierde y este es eliminado.

Ver ejemplo en figura 2 de anexos.

El nodo posee dos nodos hijo

Este caso es el más complejo con el cual lidiar al eliminar nodos en un árbol binario de búsqueda. Como el nodo cuenta con dos hijos es importante no perder sus referencias y mantener el orden. Para ello se realiza un swap entre el nodo a eliminar y el nodo más pequeño (más a la izquierda) del subárbol cuya raíz es el nodo hijo derecho del nodo a eliminar o bien el nodo más grande (más a la derecha) del subárbol cuya raíz es el nodo hijo izquierdo.

Ver ejemplo en figura 3 de anexos.

Código explicado

```
//Función principal que llama a deleteRec()
void deleteKey(int key)
{
    root = deleteRec(root, key);
}

/* A recursive function to insert a new key in BST */
Node deleteRec(Node root, int key)
{
    /* Base Case: If the tree is empty */
    if (root == null) return root; //Se pregunta si el árbol
no cuenta con una raíz, en dicho caso, se concluye que el árbol
está vacío, y retorna null

    /* Otherwise, recur down the tree */
    if (key < root.key) //Pregunta si el elemento que se desea
eliminar es menor que el elemento en el nodo a comparar
        root.left = deleteRec(root.left, key);
    else if (key > root.key) //Pregunta si el elemento que se
desea eliminar es mayor que el elemento en el nodo a comparar

        root.right = deleteRec(root.right, key);

    // if key is same as root's key, then This is the node
    // to be deleted
    else
    {
        // node with only one child or no child
        if (root.left == null)
            return root.right;
        else if (root.right == null)
            return root.left;

        // node with two children: Get the inorder successor
(smallest
        // in the right subtree)
        root.key = minValue(root.right);

        // Delete the inorder successor
        root.right = deleteRec(root.right, root.key);
    }
}
```

```
        return root;  
    }
```

Código obtenido de [GeeksForGeeks](#). [1]

Anexos

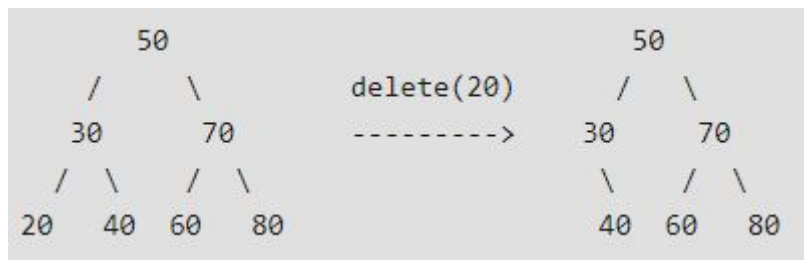


Figura 1. Ejemplo de eliminado en un nodo hoja



Figura 2. Ejemplo de eliminado en un nodo padre con un hijo.



Figura 3. Ejemplo de eliminado en un nodo padre con dos hijos.

Link al video

<https://drive.google.com/file/d/1FuOQZVfSA4HdUBzkanNf6Qc8wDHMZew7/view?usp=sharing>

Referencias

1. Geeks for Geeks. (SF). Binary Search Tree | Set 2 (Delete). Recuperado de <https://www.geeksforgeeks.org/binary-search-tree-set-2-delete/>