

Santiago Reynaldo Aguilar Vega – A01709030

Jesus Ramirez Delgado – A01274723

## **Investigación y Reflexión: BST**

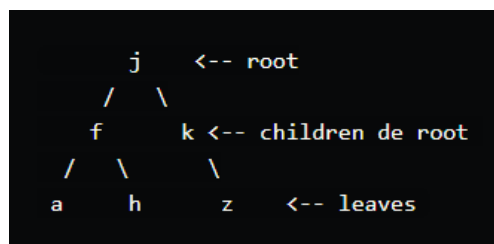
Un árbol binario de búsqueda (BST) es un tipo particular de árbol binario que presenta una estructura de datos en forma de árbol usada en informática. Un árbol binario de búsqueda cumple que el subárbol izquierdo de cualquier nodo (si no está vacío) contiene valores menores que el que contiene dicho nodo, y el subárbol derecho (si no está vacío) contiene valores mayores. Para acceder a la información de los árboles podemos dividirlo en dos secciones.

Depth-first: Pre-order, In-order, Post-order

Breadth-first: Level order

### **Vocabulario**

- El nodo principal es llamado root.
- Los nodos que están abajo directamente de un elemento son llamados children.
- Los elementos directamente de arriba son llamados parent.
- Los elementos que no tienen children son llamados leaves.



### **Usos de los BST**

- Si necesita guardar información jerárquica como por ejemplo un sistema de archivos.
- Los arboles proveen acceso moderado para búsqueda, implementando BST (Binary Search Tree), el cual es más rápido que un Linked List y más lento que un Array.
- Provee moderada inserción y eliminar elementos, el cual es más rápido que un Array y más lento que un Linked List no ordenada.

## **Tipos de BST**

Full Binary Tree: Un binary tree es un full binary tree si cada nodo tiene 0 o 2 hijos. Podemos además decir un full binary tree es un binary tree en cual todos los nodos excepto el nodo hoja tiene dos hijos.

Complete Binary Tree: Un Binary Tree puede ser Complete Binary Tree si todos los niveles están completamente llenos excepto posiblemente el ultimo nivel y el ultimo nivel tiene todas las llaves a la izquierda.

Perfect Binary Tree: Binary Tree es un Perfect Binary Tree cuando todos los nodos internos tienen dos hijos y todos los nodos hojas tienen el mismo nivel.

Balanced Binary Tree: Un Binary Tree es balanceado si la altura del árbol es  $O(\log n)$  donde  $n$  es el numero de nodos. Por ejemplo AVL

A degenerate (or pathological) tree: Es un tree donde todos los nodos internos tienen un hijo (child). Se podría decir que son iguales en rendimiento que una Linked List.

## **Reflexión**

### **Santiago:**

Considero que los arboles binarios de busqueda son una estructura de datos fundamental, ya que tiene cuenta con una facilidad para poder manipular sus nodos y poder identificarlos. Considero que ejercen mas ventajas comparados con sus

homologos los arboles binarios simples. Los usos que se le pueden implementar en la vida real son bastos como por ejemplo la organizacion de carpetas y subcarpetas. El hacer esta actividad evidencia nos ayudo a mejorar nuestras competencias y saber implementar un arbol binario, un estructura fundamental.

### **Jesus:**

Los árboles binarios de búsqueda, en lo personal ofrecen más ventajas en el momento de manipular sus nodos, como buscar, agregar y remover. Los árboles de búsqueda binarios se pueden implementar en muchas cosas, como por ejemplo en una red social, los nodos pueden ser los perfiles de lo usuarios con su información y se pueden enlazar por medio de los edges. La implementación de esta actividad en lo personal me ayudo a comprenderlos mejor, aunque creo que fue un poco confuso y lo sobre complican, más de lo que debería de ser.

### Referencias:

Geeks For Geeks. (2022). *Binary Search Tree*. Geeks For Geeks.

<https://practice.geeksforgeeks.org/batch/cpp-foundation/track/foundation-cpp-oops/video/ODY1>

Wikipedia. (2022). *Binary Search Tree*. Wikipedia.

[https://en.wikipedia.org/wiki/Binary\\_search\\_tree](https://en.wikipedia.org/wiki/Binary_search_tree)