

Arboles

Un árbol se puede definir como una estructura jerárquica y en forma no lineal, aplicada sobre una colección de elementos u objetos llamados nodos. Los árboles son considerados las estructuras de datos no lineales y dinámicas de datos muy importantes del área de computación. Los árboles son muy utilizados en informática como un método eficiente para búsquedas grandes y complejas. Casi todos los sistemas operativos almacenan sus archivos en árboles o estructuras similares a árboles. Se les llama estructuras dinámicas, porque las mismas pueden cambiar tanto de forma como de tamaño durante la ejecución del programa. Y estructuras no lineales porque cada elemento del árbol puede tener más de un sucesor.

Nodos

Se le llama nodo a cada elemento que contiene el árbol

Tipos de nodos

Nodo padre: Se utiliza este término para llamar a todos aquellos nodos que tienen al menos un hijo.

Nodo hijo: Los hijos son todos aquellos nodos que tienen un padre.

Nodo hermano: Los nodos hermanos son aquellos nodos que comparten un mismo padre en común dentro de la estructura.

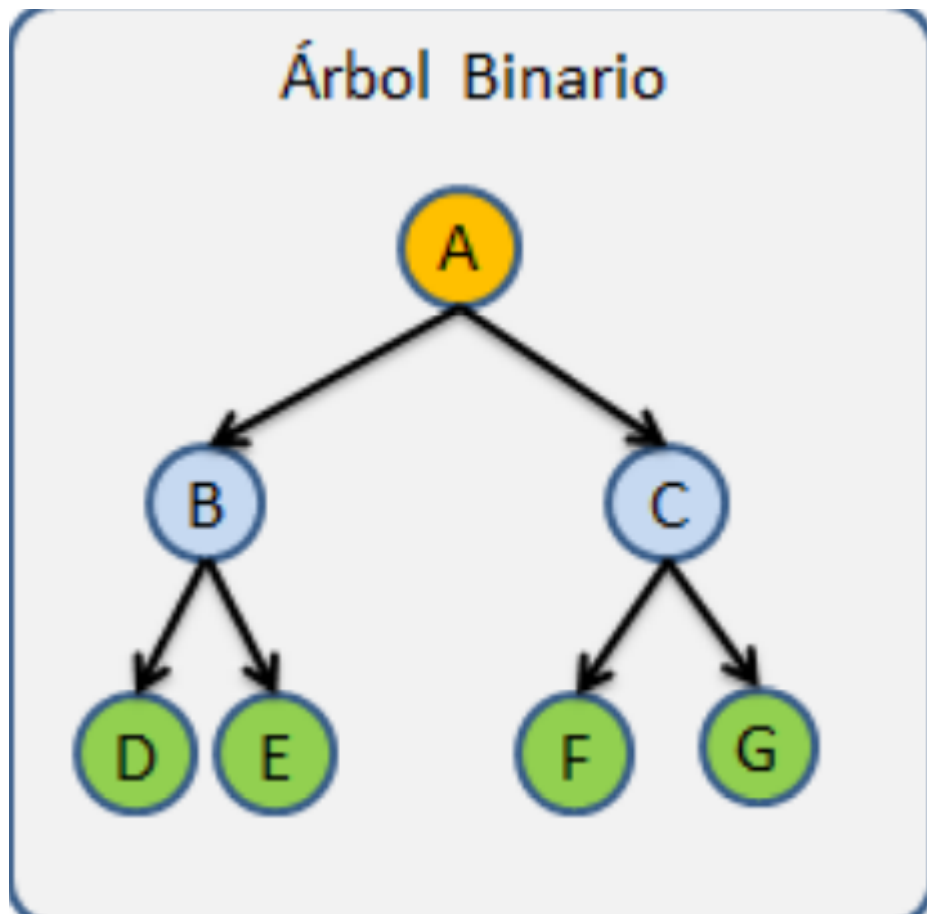
Nodo Raíz: Se refiere al primer nodo de un árbol, Solo un nodo del árbol puede ser la raíz.

Nodo Hoja: Son todos aquellos nodos que no tienen hijos, los cuales siempre se encuentran en los extremos de la estructura.

Nodo Interior o Rama: Estos son todos aquellos nodos que no son la raíz y que además tiene al menos un hijo.

Arboles binarios

Esta estructura se caracteriza por que cada nodo solo puede tener máximo 2 hijos, dicho de otra manera, es un árbol grado dos.



Tipos de arboles binarios

Árboles binarios distintos. Dos árboles binarios son distintos cuando sus estructuras son diferentes.

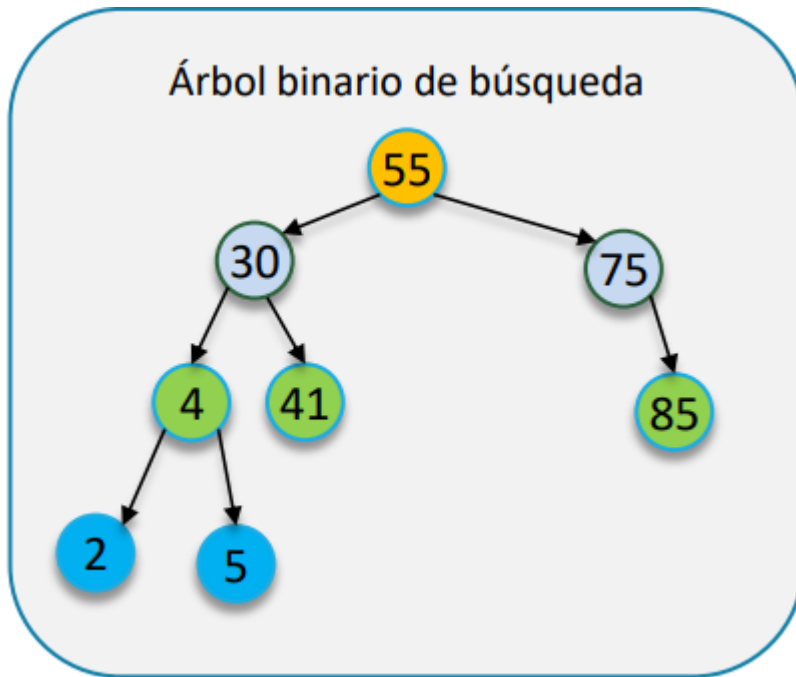
Árboles binarios similares. Dos árboles binarios son similares cuando sus estructuras son idénticas, pero la información que contienen sus nodos difiere entre sí.

Árboles binarios equivalentes. Los árboles binarios equivalentes se definen como aquellos que son similares y además los nodos contienen la misma información.

The Search Tree ADT—Binary Search Trees

Con dudas

un árbol binario de búsqueda es aquel en que, dado un nodo, todos los datos del subárbol izquierdo son menores que los datos de ese nodo, mientras que todos los datos del subárbol derecho son mayores que el nodo.



FindMin and FindMax: Estas rutinas privadas devuelven un puntero al nodo que contiene el menor y el mayor elementos del árbol, respectivamente.

Arboles AVL

es un árbol binario de búsqueda (ABB) en el que las alturas de los dos subárboles de cada nodo difieren a lo más en 1. El balance de un nodo en un árbol binario en general y de un árbol AVL en particular, se define como la altura de su subárbol izquierdo menos la altura de su subárbol derecho: $| \text{altura}(\text{arbolIzquierdo}) - \text{altura}(\text{arbolDerecho}) | < 2$

Rotaciones

es una modificación sencilla sobre la estructura de un árbol binario de búsqueda, que permite mantener la propiedad de orden.

Rotacion Simple: Rotación simple izquierda es cuando un nodo está cargado a la derecha, pero no está cargado a la izquierda. Rotación simple derecha es cuando un nodo está cargado a la izquierda, pero no está cargado a la derecha.

Rotacion Doble: es un proceso un poco más elaborado que el de la rotación simple, ya que como su nombre lo indica, implica dos rotaciones.

Conclusion

En conclusion, este es uno de los temas mas dificil de comprender en base a la programacion, debido a codigos muy largos, ademas de que cuentan con varios temas parecidos y otros que son importantes dentro de los arcoles, como los **nodods**, pero con mucha practica es posible dominar este tema, los arboles sirven para hacer ordenamientos o son estructuras jerarquicas que nos ayudan a llamar informacion, su forma de ordenamiento es la mas eficiente que existe, el tema que mas se me dificulta entender **"The Search Tree ADT —Binary Search Trees"**, debido a que no entiendo mucho la diferencia con otros tipos de arboles, tambien el **"Arbol ALV"** no lo comprendo mucho, ya que no entiendo como funciona una rotacion, los arboles son tema

largo pero util que nos puede ayudar en el futuro, a pesar de que este tema ya lo habia visto anteriormente, todavia me cuesta entenderlo.

Referencias

O, C., S, G., P.J, D., & L, J. (s.f.). Estructura de datos.

<https://www.uv.mx/personal/ermeneses/files/2021/08/Clase8-Arboles.pdf>

Árboles binarios balanceados (AVL). (s.f.). Fundamentos de la Programación Orientada a Objetos.

<https://fundamentospoorrr.blogspot.com/2017/06/arboles-binarios-balanceados-avl.html>

La creación de códigos para una estructura de datos es mucho más efectiva con un árbol AVL. (s.f.). Delfino.cr.

<https://delfino.cr/2023/06/la-creacion-de-codigos-para-una-estructura-de-datos-es-mucho-mas-efectiva-con-un-arbol-avl>

Árboles binarios balanceados (AVL). (s.f.-b). Fundamentos de la Programación Orientada a Objetos.

<https://fundamentospoorrr.blogspot.com/2017/06/arboles-binarios-balanceados-avl.html>