

Miércoles 15 de abril del 2020

UNIVERSIDAD TECNOLÓGICA DE TULANCINGO

Tecnologías de la Información y Comunicación

MATERIA:

Estructuras de datos

ARBOLES

PROFESORA:

Elizabeth García Urbina

ALUMNO:

Juárez Munguia Andrea Sarai

1718110394

Grupo:

TIC 41

Fecha de entrega:

15 de abril del 2020



**U
T
E
C**

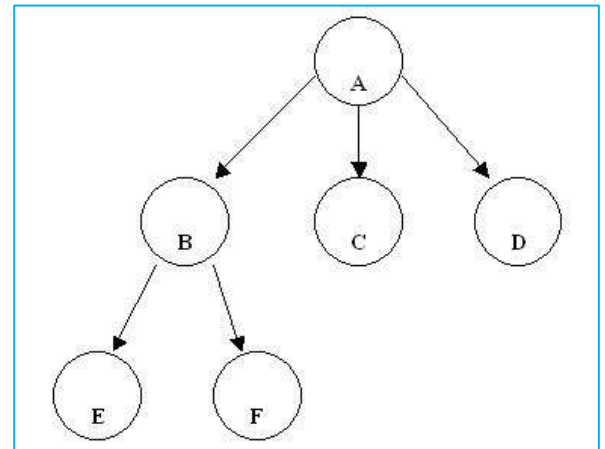
Contenido

¿QUE SON?	3
FUNCIONAMIENTO	3
OTROS DATOS.....	4
TIPOS.....	4
OPERACIONES.....	5
FUENTES	5
JUSTIFICACION DE LA PROBLEMÁTICA PARA EL PROGRAMA	6
PROGRAMA arbol.py.....	6
PROGRAMA arbol_binario.py	8

¿QUE SON?

Como sabemos los árboles entran en el tema de estructuras de datos, son una estructura de datos no lineal que se emplean en la informática. Los árboles son un poco similares a las listas doblemente enlazadas, ya que al igual que estas tienden a apuntar a otros elementos, solo que no es lineal. Si no que su estructura lógica es ramificada.

Cada uno de sus elementos se denominan nodos, y cada uno de estos nodos contienen un valor.

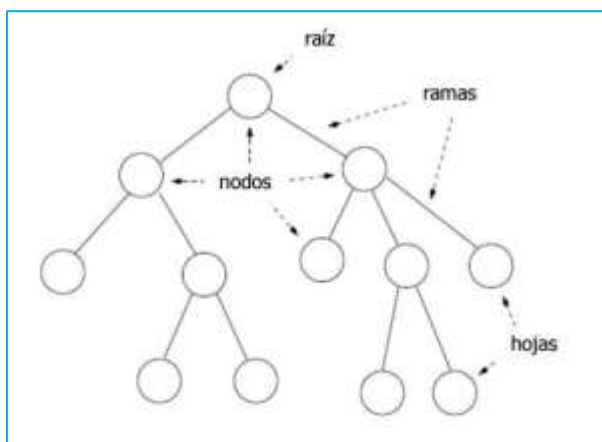


FUNCIONAMIENTO

Un árbol es una estructura de datos enlazada que organiza sus elementos en forma jerárquica. Es decir, hay una relación padre/hijos. Cada uno de sus elementos se denominan nodo u hoja, que contienen un valor. Cada nodo puede tener más de un hijo, pero solamente un padre, existe un nodo que no tiene padre al cual se le denomina raíz. Y a aquellos nodos que no tienen hijos se les denomina hojas.

Existe otro concepto que es el de profundidad, que es la distancia desde la raíz hasta la hoja más lejana.

Algunas definiciones que podrían ser útiles para una mejor comprensión son las siguientes....



- ✚ **Nodo:** Cada elemento que contiene el árbol
- ✚ **Nodo raíz:** Es el primer nodo de un árbol, solo puede hacer uno
- ✚ **Nodo padre:** Son todos aquellos nodos que tienen al menos un hijo
- ✚ **Nodo hijo:** Todos aquellos que tienen un padre
- ✚ **Nodo hermano:** Son aquellos nodos que comparten a un mismo padre dentro de la estructura
- ✚ **Nodo hoja:** Son todos aquellos nodos que no tienen hijos, siempre están en los extremos de la estructura

OTROS DATOS

Descendientes: Los descendientes de un nodo son aquellos nodos accesibles por un camino que comience en el nodo.




Ascendientes: Los ascendientes de un nodo son los nodos del camino que va desde la raíz a el nodo.

Altura de un nodo: Se define la altura de un nodo en un árbol como la longitud del camino más largo que comienza en el nodo y termina en una hoja. La altura de un nodo hoja es 0.

Altura de un árbol: se define como la altura de la raíz







Profundidad: Se define la profundidad de un nodo en un árbol como la longitud del camino que comienza en la raíz y termina en el nodo.

RECORRIDOS EN ÁRBOLES:

-  **Preorden:** Pasa por la raíz y recorre en preorden cada uno de los subárboles
-  **Postorden:** Recorre en postorden cada uno de los subárboles y pasa por la raíz
-  **Inorden:** Recorre en inorden el primer subárbol si es que existe, pasa por la raíz y al final recorre en inorden cada uno de los subárboles restantes. (Usado principalmente en árboles binarios)

TIPOS

Árboles Binarios:

1. Cada uno de sus nodos puede tener solamente 0,1 o 2 subárboles
2. Cada uno de subárboles reciben el nombre de subárbol izquierdo y subárbol derecho
3. Dentro de este tipo de árbol hay otras **variantes**:
 -  Árboles Similares, son aquellos que tienen la misma estructura
 -  Árboles Equivalentes, además de ser similares contienen la misma información
 -  Árbol Equilibrado, en este la altura de los dos subárboles de la raíz se diferencian en como máximo una unidad
 -  Árbol Completo, todos sus nodos tienen exactamente 0 o dos subárboles
 -  Árbol Lleno, en este todos los niveles están llenos
 -  Árbol Degenerado, todos sus nodos tienen solamente un subárbol

Árboles Binarios de Búsqueda:

1. Estructura donde los elementos pueden ser eficazmente localizados, insertados o borrados
2. Es ideal para manejo de gran número de operaciones
3. En cualquier nodo, todos los valores del subárbol izquierdo son menores o iguales al valor del nodo
4. Todos los valores del subárbol derecho son mayores al valor del nodo
5. Una **variante** del árbol binario de búsqueda es:
 - ✚ Árbol AVL, es balanceable es decir que la rama izquierda y la derecha no difieren en altura por más de uno.

OPERACIONES

Las inserciones se hacen en los punteros de nodos hoja o en punteros libres de nodos rama. Las operaciones que se pueden realizar dentro de la estructura de datos de árboles son:

- ✚ Añadir elementos a un árbol
- ✚ Buscar elementos dentro del árbol
- ✚ Borrar elementos existentes en el árbol
- ✚ Moverse a través del árbol por sus ramas
- ✚ Recorrer el árbol completo

FUENTES

<https://estructurasite.wordpress.com/arbol/>

<https://www.oscarblancarteblog.com/2014/08/22/estructura-de-datos-arboles/>

<https://www.infor.uva.es/~cvaca/asigs/doceda/tema4.pdf>

<https://www.fceia.unr.edu.ar/estruc/2005/arbofund.htm>

<https://es.slideshare.net/AlvaroRuano1/rboles-binarios-135045603>

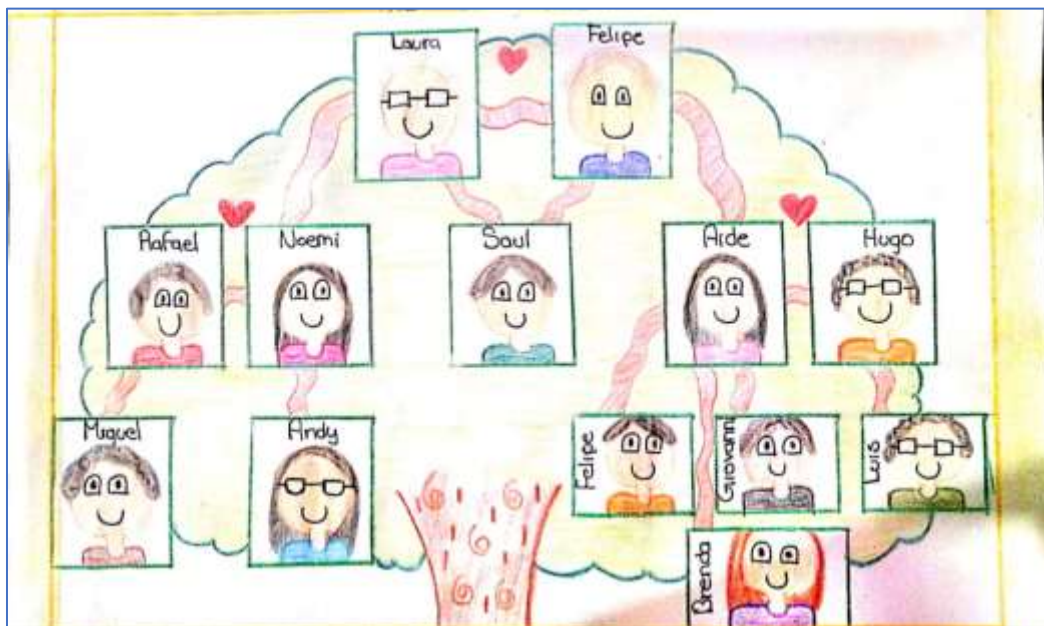
<https://runestone.academy/runestone/static/pythoned/Trees/EjemplosDeArboles.html>

<https://es.slideshare.net/AlvaroRuano1/rboles-binarios-135045603>

JUSTIFICACION DE LA PROBLEMÁTICA PARA EL PROGRAMA

PROGRAMA arbol.py

En este primer programa simplemente se muestra el concepto básico de la estructura de árbol, para que se pueda entender un poco mejor. Para ejemplificar esta estructura un claro ejemplo puede ser un árbol genealógico por la forma en que se dividen los miembros de la familia jerárquicamente. Para que se un poco más entendible, colocaré mi árbol genealógico como ejemplo.

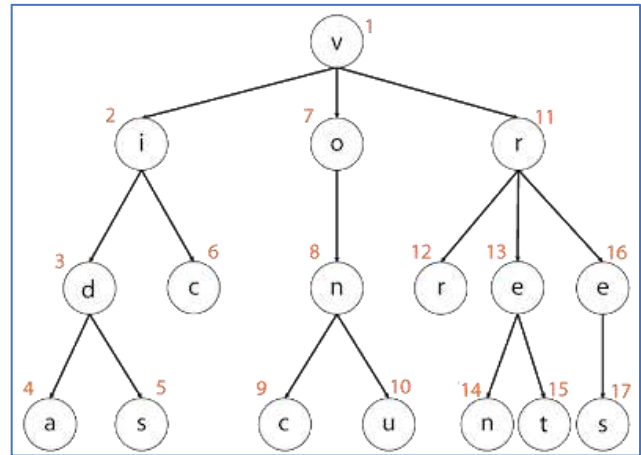


Solamente que para la implementación en el código no se esta utilizando ni el miembro Rafael, ni el miembro Hugo. Ya que son miembros un poco externos. Básicamente lo que el código permite mostrar al usuario son las dos formas de recorrido, ya sea por profundidad o por ancho.

La manera en que lo hace la forma de profundidad es que va explorando toda una rama antes de pasar a la siguiente, a continuación, una imagen y como tendría que ser la salida si el usuario elige esta opción.

De esta manera quedaría la salida de esta opción:

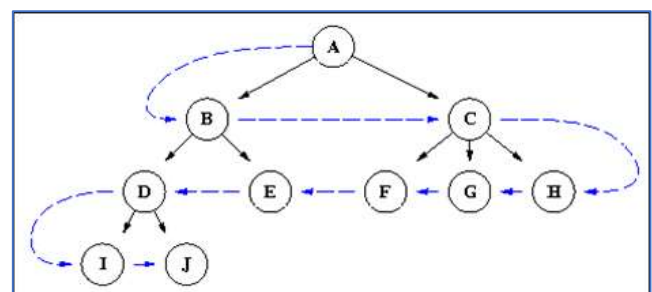
Munguia Reynoso
Noemi
Miguel
Andrea
Saul
Aide
Felipe
Giovanni
Luis
Brenda



Ahora de la forma ancho, lo que hace es que va recorriendo generación en generación (en este caso), a continuación, una imagen para ejemplificar y la salida que deberá resultar:

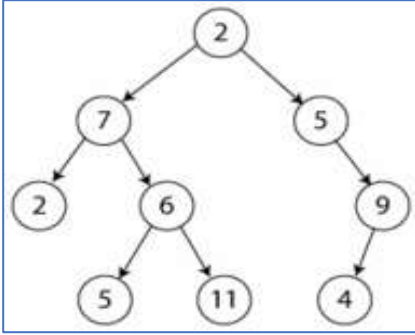
La salida al elegir la opción ancho, sería la siguiente:

Munguia Reynoso
Noemi
Saul
Aide
Miguel
Andrea
Felipe
Giovanni
Luis
Brenda



PROGRAMA arbol_binario.py

Bien para este programa decidí utilizar como problemática la clasificación de los países según el número de habitantes que cada uno de estos tiene. Lo que realiza nuestro programa es que al momento de



ingresar un nuevo elemento nos va a solicitar tanto el nombre del país como el número de habitantes y lo concatenará para que al momento que se imprima sea más entendible para el usuario. La parte donde se dará forma o sentido a la estructura de árboles binarios es con ayuda del número de habitantes, estos serán evaluados. El primero en ser ingresado será denominado la raíz y de ahí en adelante se evaluarán y compararán para saber a qué subárbol se debe agregar; si al subárbol izquierdo o al subárbol derecho.