Laboratorio Nro. 5: Árboles binarios

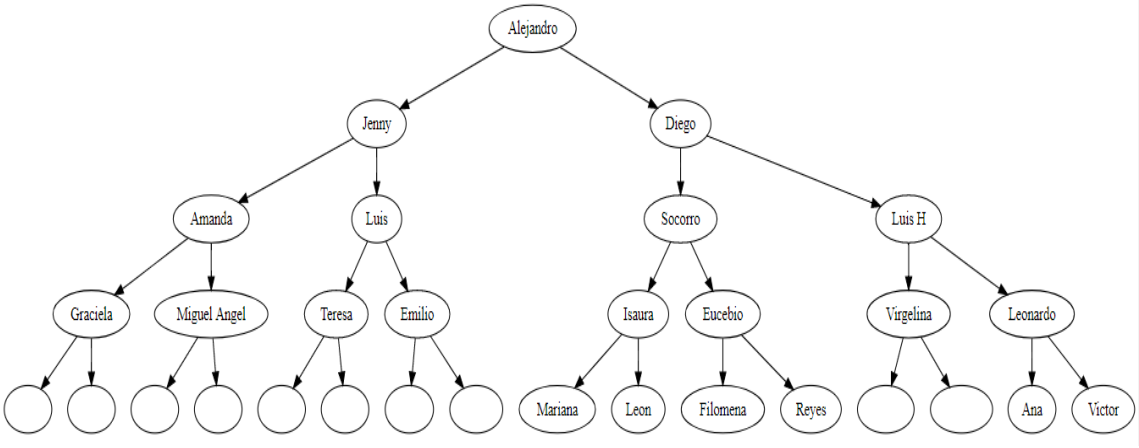
Alejandro Cano Munera
Universidad Eafit
Medellín, Colombia
acanom@eafit.edu.co

Jorge Luis Herrera Chamat
Universidad Eafit
Medellín, Colombia
jlherrerac@eafit.edu.co

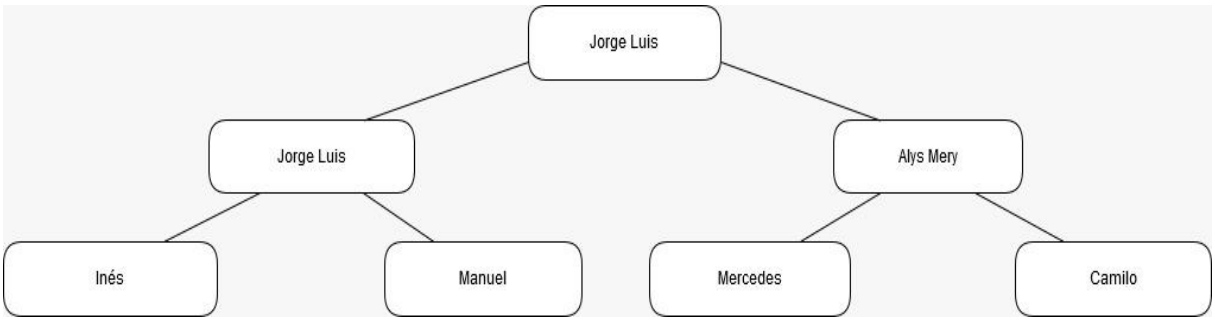
3) Simulacro de preguntas de sustentación de Proyectos

1. Árboles Genealógicos:

Alejandro Cano Munera:



Jorge Luis Herrera Chamat:



2. El árbol genealógico genérico de descendientes directos funciona como un árbol binario, siendo la raíz del mismo la persona desde la cual se va a iniciar el mapeo de ancestros y los dos nodos hijos de cada nodo son los padres del nodo que se está evaluando. Conservando este modelo de árbol binario se mantiene información importante sobre el orden de descendencia en la línea genealógica (padres – hijo), sin embargo, a la hora de buscar a alguna persona, en el peor de los casos tendremos una complejidad de O(n). Por otra parte, si se pretendería crear un modelo de árbol AVL donde los nodos se auto equilibran, posiblemente se perdería información sobre el orden de la descendencia, y además seguiríamos teniendo una complejidad de O(n) al intentar buscar a alguien, ya que para esto deberíamos empezar a recorrer todo el árbol hasta encontrar o verificar que no está una determinada persona. En conclusión, para un árbol genealógico la inserción y la búsqueda de una determinada persona en el peor de los casos deberá ser O(n).

3. Pos-orden: En este ejercicio primero se leen todos los números de entrada (que se encuentran en pre-orden) y se agregan a un árbol binario haciendo uso del método insertar(), se sabe que se han termino de ingresar todos los datos cuando se da un enter “vacío” o sin caracteres, luego se procede a recorrer e imprimir dicho árbol en pos-orden, haciendo uso del método printPosorden(), de la siguiente manera: Nodo izquierdo, nodo derecho y raíz. Adicionalmente el código cuenta con el manejo de excepciones, por ejemplo, cuando se intenta añadir un carácter que no es de tipo numérico.

   TreeSumming: El ejercicio lee los datos por línea y se posiciona en cada nodo para crear los nodos del siguiente nivel. Después de generar el árbol, luego se leen todas las ramas con recursión hasta encontrar una rama que cumpla la condición, o hasta llegar al final del árbol en caso de que no sea posible cumplir la condición.

4. Pos-orden: $T(n) = C1*n*n + C2*n + C3$ es $O(n^2)$

   TreeSumming: $T(n) = n*C1 + C2$ es $O(n)$

5. Pos-orden: En este ejercicio n significa el número de elementos o enteros que se ingresan y se deberán imprimir en pos-orden.

   TreeSumming: En este ejercicio n significa la cantidad total de elementos que se almacenaran en los nodos del árbol.

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

4) Simulacro de Parcial

1. **a.** altura(raiz.izq) + 1
   b. altura(raiz.der) + 1

2. C

3. **a.** false
   b. a.dato
   c. a.izq, suma - a.dato
   d. a.der, suma - a.dato

4. **4.1.** C
   4.2. A
   4.3. D (Wilkenson, Joaquina, Estaquia, Florinda, Eustaquio, Jovin, Sufranio, Piolina, Wilberta, Piolín, Usnavy)

   4.4. A (Usnavy, piolin, Wilberta, Piolina, Sufranio, Jovin, Eustaquio, Florinda, Eustaquia, Joaquina, Wilkenson)

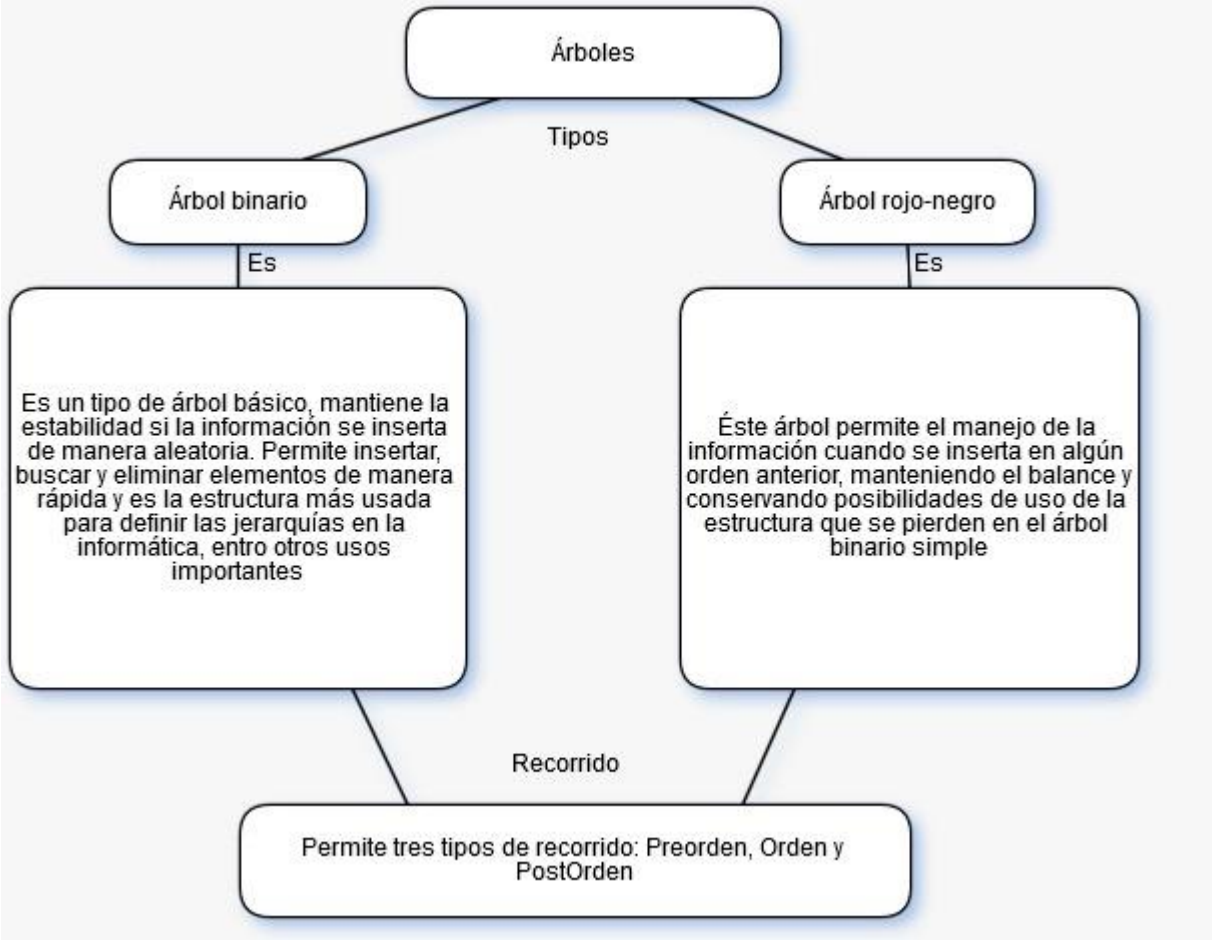
5. **a.** toInsert == p.data
   b. toInsert > p.data

5. Lectura recomendada:

a) Título: Chapter 8: Binary trees

b) Resumen: Los árboles son estructuras que combinan la eficiencia de otras dos estructuras: arreglos y listas enlazadas. Se puede buscar, insertar y eliminar elementos de manera rápida. Éstos han sido estudiados como entidades matemáticas abstractas, por lo que hay un conocimiento teórico muy extenso sobre ellos. Los más simples, llamados árboles binarios, sólo son eficientes si la información se inserta de forma aleatoria, pero si se inserta en algún tipo de orden el árbol se vuelve desbalanceado, para este tipo de operaciones se utiliza el árbol rojo-negro, que permite mantener el balance Los árboles permiten recorrer los datos de tres maneras diferentes: Preorden, Inorden y postorden. Estos métodos permiten utilizar la información de una u otra manera dependiendo de la necesidad que se tenga, logrando una mayor eficiencia a la hora de utilizar esta estructura de datos.

c) Mapa de Conceptos:



DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co