Laboratorio Nro. 5: Árboles binarios

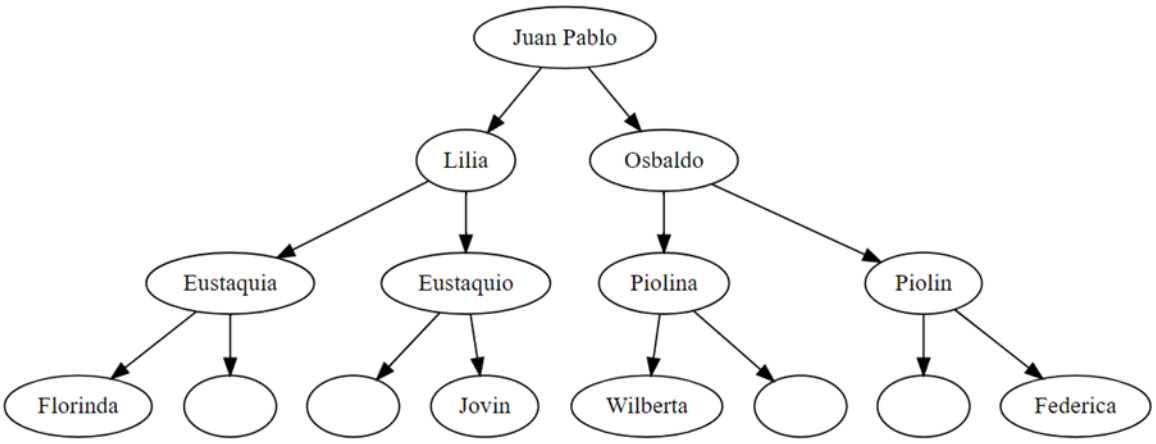
Alejandro Murillo González
Universidad Eafit
Medellín, Colombia
amurillog@eafit.edu.co

Juan Pablo Vidal Correa
Universidad Eafit
Medellín, Colombia
Jpvidalc@ eafit.edu.co

3) *3) Simulacro de preguntas de sustentación de Proyectos*

3.1 Investiguen cuáles son sus nombres, los nombres de sus padres, los nombres de sus abuelos y los nombres de sus bisabuelos, y construyan sus árboles genealógicos.

Grafico:



Código:

Nota: las clases Node y BinaryTree están definidas en el repositorio de GitHub, en el taller 10.

```
import Node
import BinaryTree

#Left Subtree
florinda = Node.Node("Florinda")
eustaquia = Node.Node("Eustaquia", florinda)
jovin = Node.Node("Jovin")
```

```
eustaquio = Node.Node("Eustaquio", right_node = jovin)
Lilia = Node.Node("Lilia", eustaquia, eustaquio)

#Right Subtree
wilberta = Node.Node("Wilberta")
piolina = Node.Node("Piolina", wilberta)
Federica = Node.Node("Federica")
piolin = Node.Node("Piolin", right_node = Federica)
Osbaldo = Node.Node("Osbaldo", piolina, piolin)

#Root
root = Node.Node("Juan Pablo", Lilia, Osbaldo)

tree = BinaryTree.BinaryTree(root)

# --- BFS Tree traversal
print("\n--- BFS Tree traversal ---")
tree.bfs_traverse()

# --- DFS Tree traversal
print("\n--- DFS Tree traversal ---")
tree.dfs_traverse()

# --- Drawing Tree
print("\n--- Drawn tree at: http://webgraphviz.com/ ---")
print(tree.draw_tree())
```

3.2 ¿Se puede implementar más eficientemente un árbol genealógico para que la búsqueda e inserción se puedan hacer en tiempo logarítmico? ¿O no se puede?

No, ya que para la inserción se necesita una complejidad de O(n), ya que hay que balancear el árbol, porque se le debe asignar un peso específico a cada individuo para que se reconozca el grado de familiaridad con el individuo de la raíz y el género (padres y madres). Además para obtener una búsqueda eficiente se necesita de una inserción con una complejidad capaz de organizar meticulosamente los lazos entre familia.

3.3 Expliquen con sus propias palabras cómo funcionan los ejercicios del numeral 2.1

Para el ejercicio propuesto hay que pasar de un recorrido pre-orden a uno pos-orden, para ello se hace uso de una deque o cola de doble extremo, la cual combina los elementos de una pila y una cola, lo que permite agregar y

eliminar elementos al principio o al final, por lo que agregando los elementos en pre-orden, se pueden ir sacando en la forma pos-orden.

3.4 Calculen la complejidad de los ejercicios realizados en los numerales 2.1 y 2.2 y agréguenla al informe PDF

La complejidad de este ejercicio es O(n), ya que eligiendo el peor de los casos solo tiene un ciclo.

Código:

```
def preorder_to_postorder():

 node = input()
 tree = BinaryTree.BinaryTree(BinaryTreeNode.BinaryTreeNode(int(node)))
 while node != "":                                                    //C*n
    tree.insert(BinaryTreeNode.BinaryTreeNode(int(node)))
   node = input()

 fringe = deque([tree.root])
   postorder = deque([])
 while len(fringe) > 0:                                               //C*n
   next_node = fringe.popleft()
    postorder.appendleft(next_node)

   if next_node.left_node is not None:                                //C*n
    fringe.appendleft(next_node.left_node)

    if next_node.right_node is not None:                              //C*n
  fringe.appendleft(next_node.right_node)
```

```
for node in postorder:                                    //O(1)
  print(node)
```

T(n) = O(C x n + C x n + 1) == T(n) es O(n)

3.5 Expliquen con sus palabras las variables (qué es 'n', qué es 'm', etc.) del cálculo de complejidad del numeral 3.3

Las variables "n", "m" y/o "y" son las variables de entrada, esto con el objetivo de facilitar los cálculos de complejidad.

4) Simulacro de Parcial

1.

a) altura(raiz.izquierda)+1

b) altura(raiz.derecha)+1

2. C

3.

a) false

b) a

c) a.izq +1 , suma + a

c) a.der -1 , suma – a

4.

4.1) C
4.2) A
4.3) D
4.4) A

5.

a) toInsert = 0

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

b) p < toInsert

5) Lectura recomendada (opcional)

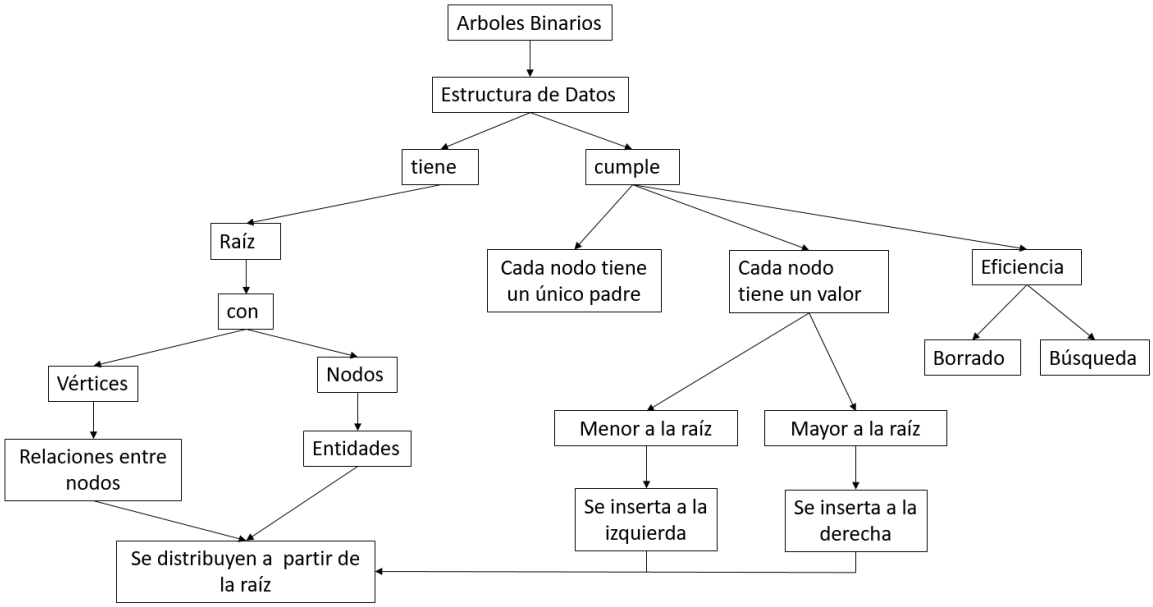
a) Resumen:

Arboles Binarios:

Los arboles binarios son un tipo de estructura de datos que ofrecen una búsqueda y un borrado muy eficientes. En estos existe una raíz la cual es un nodo del que salen vértices y otros nodos, cada nodo solo puede tener un padre pero puede tener múltiples hijos. Los nodos son la representación de entidades como personas, carros, parqueaderos, entre otros; por otro los vértices son las relaciones entre cada nodo. Por lo que para recorrer el árbol se necesita seguir el patrón de los nodos y vértices, lo cual se puede hacer siguiendo órdenes específicos que pueden reducir el tiempo de búsqueda considerablemente. Para esto se tienen recorridos como pos-orden, el cual recorre el árbol de izquierda a derecha, y por ultimo pasa por la raíz; se tiene el indorden, el cual va de izquierda a la raíz y luego a la derecha; y también está el recorrido pre-orden, el cual empieza en la raíz, y va de izquierda a derecha.

Para crear un árbol se debe crear primero la raíz, de la cual se va generando cada nodo, buscando el mejor lugar para su inserción, para que el patrón tenga un recorrido eficiente, para hacer esto se le entrega un valor a cada nodo, y dependiendo su tamaño se distribuye en la izquierda o la derecha; si el valor es mayor al de la raíz este va en la derecha del árbol, y si es menor va a la izquierda, en la mayoría de los casos se debe hacer un balance en el árbol para distribuir bien estos nodos y así no queden concentrados en un mismo lado.

b) Mapa conceptual:



6) Trabajo en Equipo y Progreso Gradual (Opcional)

a) Actas de reunión:

Integrante	Fecha	Hecho	Haciendo	Por Hacer
Murillo	10/10/2017	Implemento Binary tree en python	Árbol genealógico	
Vidal	12/10/2017	Implemento código en línea (2.1)	Calculando complejidad	leer lectura opcional y resumen

Murillo	13/10/2017	Código numeral 1	Organizar texto- informe de laboratorio	Parte 3 del laboratorio
Vidal	18/09/2017	implemento resumen	Simulacro de parcial	Organizar texto- Informe de laboratorio

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