Laboratorio Nro. 4 Árboles binarios

Jamer José Rebolledo Quiroz
Universidad EAFIT
Medellín, Colombia
jjrebolleq@eafit.edu.co

Andrés Felipe Tamayo Arango
Universidad EAFIT
Medellín, Colombia
aftamayoa@eafit.edu.co

23 de abril de 2018

3) Simulacro de preguntas de sustentación de Proyectos

3.1 ¿Se puede implementar más eficientemente un árbol genealógico para que la búsqueda e inserción se puedan hacer en tiempo logarítmico? ¿O no se puede? ¿Por qué?
R. No se puede, al ser una familia también se tiene que buscar miembro por miembro.
3.2 Expliquen con sus propias palabras cómo funcionan los ejercicios de los numerales 2.1 y 2.2
R. En el ejercicio 2.1 por medio de Scanner, se leen los elementos de un árbol por consola, luego con una función de llamado recursivo, mandábamos a imprimir primero el nodo izquierdo, luego el derecho y por último la raíz.
3.3 Calculen la complejidad de los ejercicios realizados en los numerales 2.1 y 2.2

```
private void imprimirPos(Nodo nodo){
        if(nodo.padre == raiz){}
        else
        if(nodo != null){
            if(nodo.izq != null)System.out.println(nodo.izq.dato); \\ C1
            if(nodo.der != null)System.out.println(nodo.der.dato); \\ C2
            if(nodo.padre != null){
                imprimirPos(nodo.padre); \\ T(n - 1)
            }
        }
    }
```

Complejidad = $O(n)$
3.4 Expliquen con sus palabras las variables (qué es n, qué es m, etc.) del cálculo de complejidad del numeral 3.4
R. n es el la distancia del nodo en que está parado el método a la raíz.

4) Simulacro de parcial en el informe PDF

1. El nodo de un árbol binario se define así:

```
class Nodo {
            Nodo izq;
            Nodo der;
            int dato;
      }
```

Kefo ha dañado a Dayla su código para determinar cuál es la altura máxima de un árbol binario. Dayla no recuerda como lo había hecho y les pide ayuda para que lo completen:

```
int altura(Nodo raiz){
                if(raiz == null)
                return 0;
                int izq =_______;
                int der =_______;
                return Math.max(izq, der);}
```

a) Completen el espacio en línea 04
Respuesta. altura(raiz.izq);

b) Completen el espacio en línea 05
Respuesta. altura(raiz.der);

2. A un árbol binario de búsqueda se le ingresan 9 elementos en este orden: 12, 3, 5, 8, 2, 1, 9, 17. £Cuántos nodos hay que recorrer antes de encontrar el número 8? Se cuenta la raíz, pero no se cuenta el nodo donde está el 8.

a) 2

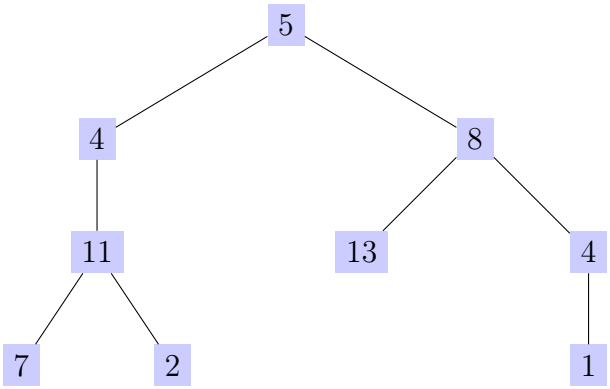
b) 1

c) 3

d) 5

Respuesta. b) 3

3. Definimos el camino desde la raíz hasta una hoja en un árbol binario como una secuencia de nodos empezando en el nodo raíz y bajando hasta una hoja. Una hoja es un nodo que no tiene hijos. Decimos que un árbol vacío no contiene caminos desde la raíz hasta una hoja. Como un ejemplo, el siguiente árbol tiene 4 caminos desde la raíz hasta una hoja:



Camino desde la raíz hasta la hoja:

- **Camino 1:** 5 4 11 7
- **Camino 2:** 5 4 11 2
- **Camino 3:** 5 8 13
- **Camino 4:** 5 8 4 1

Para este problema nos interesan las sumas de los elementos en los nodos de esos caminos, por ejemplo, la suma del camino [5,4,11,7] es 27.
Dada la raíz de un árbol binario Nodo a y un entero int suma, decir si existe un camino desde la raíz hasta una hoja tal que al sumar los valores de los nodos de ese camino la suma sea igual al parámetro suma.
Retorne falso si no se puede encontrar un camino con esa condición. Utilice la definición de Nodo del punto 1.
Desafortunadamente, al código que hizo Dayla le faltan unas líneas y Kefo está de vacaciones.

```
1 boolean sumaElCamino(Nodo a, int suma) {
2         if (a == null)
3         return __________;
4         if (a.izq == null && a.der == null)
5         return suma == __________;
6         else
7         return sumaElCamino(_______, _________)
8         || sumaElCamino(_______, _________); }
```

a) Completen el espacio de la línea 03
Respuesta. return false;

b) Completen el espacio de la línea 05
Respuesta. return suma == a.dato;

c) Complete los espacios de la línea 07
Respuesta. return sumaElcamino(a.izq, suma - a.dato);

Docente Mauricio Toro Bermudez
Celular: (+57)(4)2619500 Ext. 9473. Oficina: 19 – 627
E-mail: mtorobe@eafit.edu.co

d) Complete los espacios de la línea 08
Respuesta. return sumaElcamino(a.der, suma - a.dato);

4. Consideren la siguiente definición de árbol binario:

```
class Node {
                public Node left;
                public Node right;
                public String data;
                public Node(String d) {
                data = d;
                }
                }
```

El siguiente algoritmo imprime todos los valores de un árbol en pre orden.

```
01      private void printAUX(Node node) {
02              if (node != null) {
03              System.out.println(node.data);
04              printAUX(node.left);
05              printAUX(node.right);
06              }
07      }
08              public boolean print() {
09              printAUX(root);
10              }
```

4.1 £Cuál ecuación de recurrencia que describe el número de instrucciones que ejecuta el algoritmo print en el peor de los casos?
La variable n representa el número de elementos del árbol.

a) $T(n) = T(n-1) + C$

b) $T(n) = 2.T(n-1) + C$

c) $T(n) = 2.T(n/2) + C$

d) $T(n) = T(n/2) + C$

e) $T(n) = T(n+1) + C$

Respuesta. $T(n) = 2.T(n/2) + C$
4.2 £Cuál es su complejidad asintática en el peor de los casos del algoritmo print? La variable n representa el número de elementos del árbol.

a) $O(n)$

b) $O(n^2)$

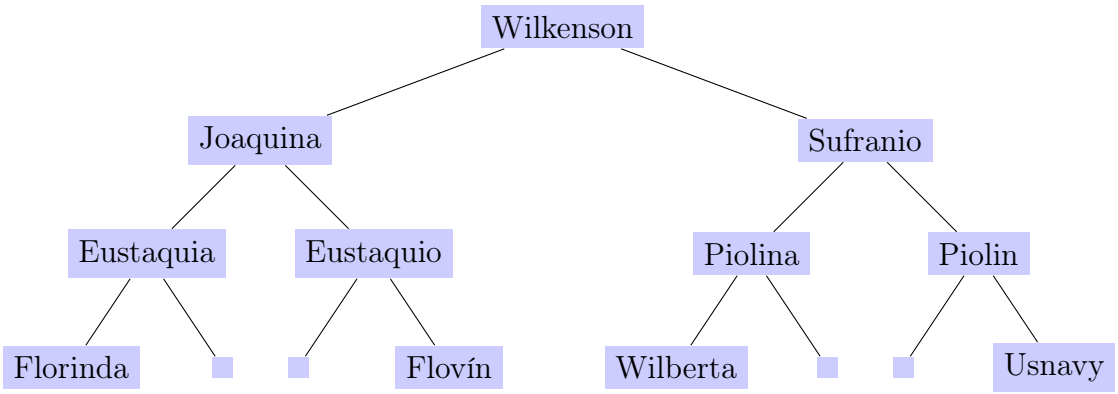
c) $O(log(n))$

d) $O(n * m)$

d) $O(1)$

Respuesta. $O(n)$

4.3 £Cuá el es la salida del algoritmo print para el siguiente árbol? Tenga en cuenta que la raíz es Wilkenson.
Después de hacer la rotación, usted entenderá que realmente las mujeres van a la izquierda y los hombres a la derecha.
Tenga en cuenta que, en un árbol genealógico, para algunas personas, no se conoce la mamá o el papá.



a) Wilkenson, Sufranio, Piolín, Usnavy, Piolina, Wilberta, Joaquina, Eustaquio, Florinda, Eustaquia, Yovín

b) Sufranio, Piolina, Wilberta, Piolín, Usnavy, Joaquina, Estaquia, Florinda, Wilkenson, Yovín, Eustaquio

c) Wilkenson, Yovín, Eustaquio, Sufranio, Piolina, Wilberta, Piolín, Usnavy, Joaquina, Estaquia, Florinda

d) Wilkenson, Joaquina, Eustaquia, Florinda, Eustaquio, Jovín, Sufranio, Piolina, Wilberta, Piolín, Usnavy

e) Sufranio, Piolina, Wilberta, Piolín, Usnavy, Florinda, Wilkenson, Yovín, Eustaquio, Joaquina, Estaquia

Respuesta. d) Wilkenson, Joaquina, Eustaquia, Florinda, Eustaquio, Jovín, Sufranio, Piolina, Wilberta, Piolín, Usnavy.

4.4 £Qué modificación hay que hacer algoritmo print para que arroje la siguiente respuesta para el árbol anterior?:
Usnavy, Piolín, Wilberta, Piolina, Sufranio, Florinda, Eustaquio, Yovín, Eustaquia, Joaquina, Wilkenson.
Tenga en cuenta que la raíz es Wilkenson. Como Wilkenson es la raíz, para poder entender el árbol, recomendamos rotarlo 180 grados. Después de hacer la rotación, usted entenderá que realmente las mujeres van a la izquierda y los hombres a la derecha.
Tenga en cuenta que, en un árbol genealógico, para algunas personas, no se conoce la mamá o el papá. ||

a) Cambiar el orden de las lineas 03, 04 y 05 por 05, 04, 03

b) Cambiar el orden de las lineas 03, 04 y 05 por 04, 05, 03

c) Cambiar el orden de las lineas 03, 04 y 05 por 03, 05, 04

d) Cambiar el orden de las lineas 03, 04 por 06, 07

e) Intercambiar la linea 03 y la linea 04

Respuesta. a) Cambiar el orden de las lineas 03, 04 y 05 por 05, 04, 03.

5 Luis escribió un programa para insertar un número en un árbol binario de búsqueda. En dicho árbol, él quiere tener los números menores o iguales a la raíz a la derecha y los mayores a la raíz a la izquierda. Ayúdele a completar su código. El algoritmo recibe la raíz de un árbol p y un número a insertar toInsert, y retorna la raíz del árbol con el elemento insertado donde corresponde.

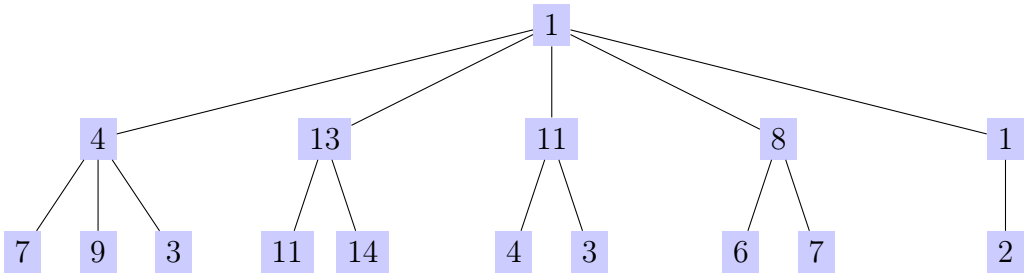
```
01      private Node insert(Node p,int toInsert){
02              if (p == null)
03              return new Node(toInsert);
04              if (...........)
05              return p;
06              if (...........)
07              p.left = insert(p.left, toInsert);
08              else
09              p.right = insert(p.right, toInsert);
10              return p;
11      }
```

a) Complete, por favor, la línea 4 con la condición que corresponde
if(p.data == toInsert)

b) Complete, por favor, la línea 6 con la condición que corresponde
if(toInsert > p.data)

6 Dado un árbol nario, un camino desde la raíz a cualquiera de sus hojas se considera simple si la suma de sus elementos pares es igual a la suma de sus elementos impares. Por ejemplo, un camino desde la raíz hasta una hoja con los elementos [1, 2, 1] es simple, pero el camino desde la raíz hasta una hoja [1, 3, 1] no es simple. Su tarea es determinar cuántos caminos simples hay en un árbol. La implementación de un nodo nario es la siguiente:

```
class NNodo{
        int val; //Valor en el nodo actual.
        //Hijos del nodo actual.
        LinkedList<NNodo> hijos;
}
```



6.1 £Cuantos caminos simples hay desde una raíz hasta una hoja en el árbol anterior?

a) 1

b) 2

c) 3

d) 4

Respuesta. d) 4
El siguiente código nos ayuda a determinar el número de caminos simples en un árbol, pero faltan algunas líneas. Por favor, complétenlas.

```
01      public int cuantosSimples(NNodo raiz, int suma){
02              //Arbol vacio
03              if(raiz == null)
04              _____________;
05              //Hoja
06              if(raiz.hijos.size() _______) // Si suma es 0
07              return (suma == 0) ? 1 : 0; // Retorne 1, sino, 0
08              int total = 0;
09              for(NNodo n: raiz.hijos)
10              if(n.val %2 == 0) // Par
11              total += cuantosSimples(n, suma + n.val);
12              else // Impar
13              total += cuantosSimples(n, suma - n.val);
15              return total;
16      }
17
18      public int cuantosSimples(NNodo raiz){
19              int val = (raiz.val %2 == 0) ?
                raiz.val : -raiz.val;
20              return cuantosSimples(raiz, val);
21      }
```

6.2 Completen la línea 4
Respuesta return 0;
6.3 Completen la línea 6
Respuesta
7. . Los siguientes algoritmos son el recorrido en in orden y pos orden. En la vida real, estos recorridos son de utilidad para hacer procesamiento del lenguaje natural.

```
void InOrden(Node node){
        if (node != null){
        InOrden(node.left);
        System.out.println(node.data);
        InOrden(node.right);
} }
void PosOrden(Node node){
        if (node != null){
        PosOrden(node.left);
        PosOrden(node.right);
        System.out.println(node.data);
} }
```

Consideren el recorrido in orden y pos orden de un Árbol binario de búsqueda con los siguientes elementos 4, 9, 1, 5, 7, 11, 13, 2, 0, 10. Ahora, la impresión del recorrido en orden nos devolverá los elementos , y, la impresión del recorrido pos-orden devolverá.
7.1 £Cuál es la impresión pos-orden?

1. 0, 2, 1, 7, 5, 10, 13, 11, 9, 4
2. 0, 1, 2, 4, 5, 7, 9, 10, 11, 13
3. 0, 2, 1, 7, 10, 5, 13, 11, 9, 4
4. 4, 1, 0, 2, 9, 5, 7, 11, 13, 10

Respuesta 1. 0, 2, 1, 7, 5, 10, 13, 11, 9, 4
7.2 Supongan que hacemos la comparación de y , £cuál es la cantidad de elementos para los cuales no es igual a ?

1. 7
2. 2
3. 5
4. 8