Laboratory practice No. 4: TREES

Juan Andres Arroyave
Universidad Eafit
Medellín, Colombia
jaarroyavs@eafit.edu.co

Gustavo A. López Garcia
Universidad Eafit
Medellín, Colombia
galopezg@eafit.edu.co

Juan Camilo Iguaran
Universidad Eafit
Medellín, Colombia
jciguaranf@eafit.edu.co

3) Practice for final project defense presentation

3.1

The type of tree is binary.

Complexity:

```
  private int max2(int i, int j)
    {
        if (i > j)//C1
            return i;//C2
        return j;//C3
    }
C3

private int maxheightAUX(Node node)
    {
        if (node == null)//C1
                return 0;//C2
        else
                return max2(maxheightAUX(node.left), maxheightAUX(node.right))+1;//C3*n
    }
//C3(n-1)

    public int maxheight()
    {
       return maxheightAUX(root);//C1
    }
```

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

3.3

The node class will be responsible for creating the 3 attributes with which the tree will be built. Create an attribute of type integer that will be the numbers entered in the tree and two attributes of type Node that will create the branches of the tree. The SearchBinaryTree class will be responsible for creating that tree, and, in addition to printing the data entered in two ways: in preorder and postorder. First, a Node class attribute named root will be created, which will be the root of the tree, then we proceed to insert each data through the nodes. We ask if the root is null, if so, then the data that we insert will go to the root, if not, a previous node that is equal to the root will be created, or, to the previous data. Within the else condition a while cycle will be done so long as the branch is not null, that is to say that it has something, keep running. If the data entered is less than the previous data, it will go to the left branch, otherwise it will go to the right and the new node will be created. Subsequently, a public method is created that will be responsible for calling a private method that will print the data in preorder. First ask if it is null, if so, then it will not return, otherwise it will make a recursive call to the method by first printing the root and recursively calling the nodes on the left and then on the right. It also happens with the public method PostOrden; call the private method auxPostOrden, recursively printing the nodes on the left, then the nodes on the right and finally the node on the root.

3.4

```
public void insertar (int data)
    {
       Nodo nodo;// C1
       nodo = new Nodo (); // C2
       nodo.data = data; //C3
       nodo.izq = null;//C3
       nodo.der = null;//C4
       if (root == null)//C5
          root = nodo;//C6
       else
       {
          Nodo anterior = null, reco; //C7
          reco = root;//C8
          while (reco != null) //C8*n
          {
             anterior = reco; //C9*n
             if (data < reco.data)//C10*n
                reco = reco.izq;//C11*n
             else
                reco = reco.der;//C12*n
          }
          if (data < anterior.data)//C13*n
             anterior.izq = nodo;//C14*n
          else
             anterior.der = nodo;//C15*n
       }
```

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

```
    }
//C16(n-1)

public void PosOtrden(){
        auxPostOrden(root);//C1
      }

private void auxPostOrden(Nodo nodo){
         if(nodo == null){//C1
            return;//C2
          }
          auxPostOrden(nodo.izq);//C3*n
          auxPostOrden(nodo.der);//C4*n
          System.out.println(nodo.data + " ");//C5
      }
//C4(n-1)
```

3.5 The variables c1, c2, c3, etc ... that appear in the calculation of complexity of the methods refer to the steps that the program does in any case or in the worst case. The variable n refers to the number of times each step is done. For example, in a cycle that is traversed 5 times, then n is equal to 5, which means that step x will be traversed n times.

4) Practice for midterms

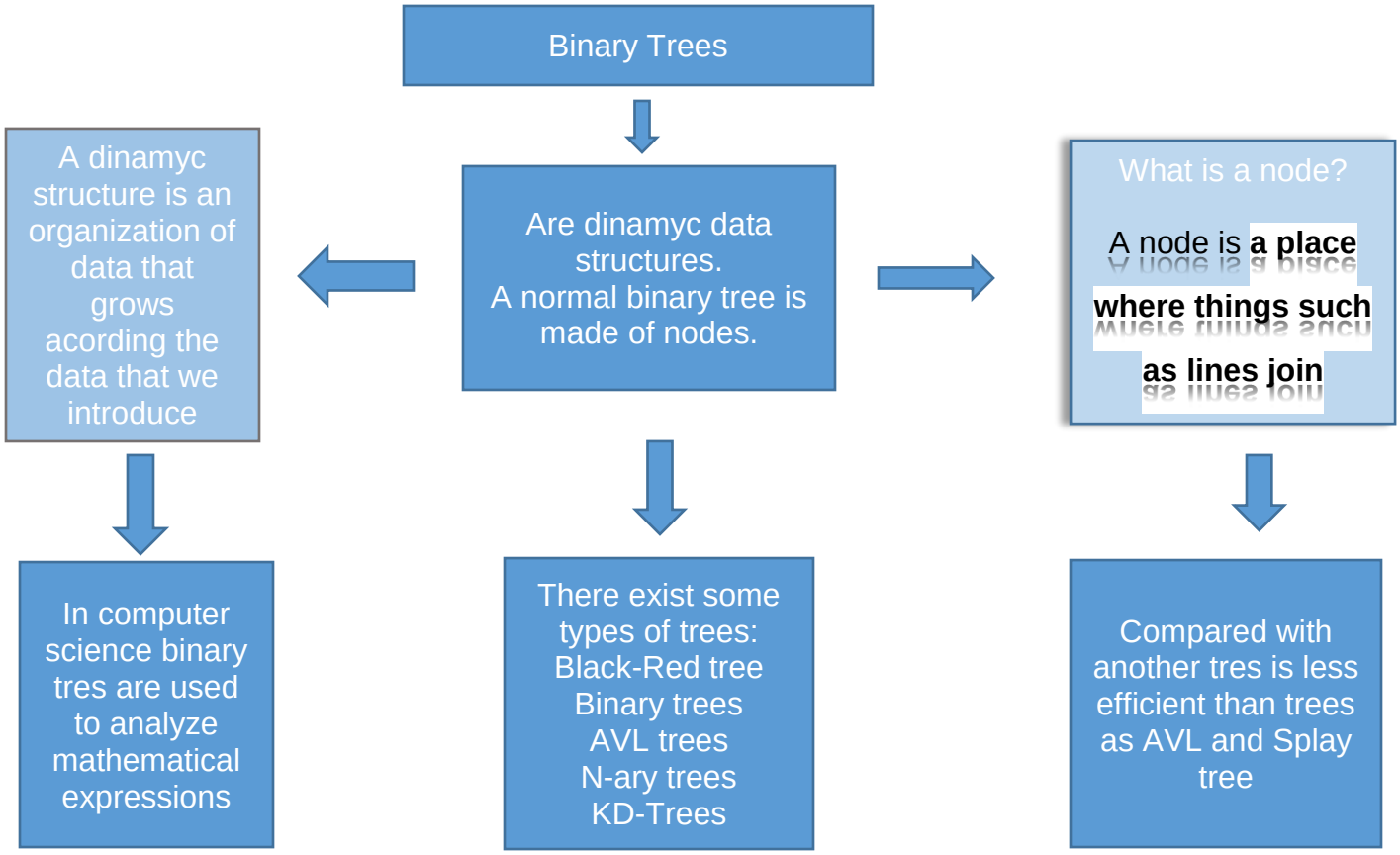
4.1)
a) raíz.izq.dato;
b) raíz.der.dato;
4.2) To find the number 8 you have to go through 3 nodes which corresponds to the answer C.
4.3)
a) 0;
b) suma+a.dato;
c) (a.der, suma)
d) (a.izq, suma)
4.4)
4.4.1) The answer that corresponds to the recurrence equation is T (N) = 2 * T (N-1) + C which equals B.
4.4.2) The recurrence equation that best describes the number of instructions executed by the algorithm is O (n ^ 2) that corresponds to B.
4.4.3) The output given by the print algorithm for this tree is Wilkenson, Joaquina, Eustaquia, Florinda, Eustaquio, Jovín, Sufranio, Piolina, Wilberta, Piolin, Usnavy corresponding to answer D.
4.4.4) To obtain the desired answer, the order of lines 03, 04 and 05 must be changed to 05, 04 and 03, corresponding to A.

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

4.5)
a) if(p != null)
b) if(p.left > toInsert)
4.7)
4.7.1) The answer corresponds to D.
4.7.2) The answer corresponds to B.
4.8) The answer corresponds to D.
4.9) The answer corresponds to B.
4.10) Yes it is a binary search tree.
4.11)
4.11.1) The answer corresponds to C.
4.11.2) The answer corresponds to B.
4.11.3) It is not a binary search tree.

5) Recommended reading (optional)

Mapa conceptual about binary trees



PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473