Laboratory practice No. 4: Trees

Santiago Cartagena Agudelo
Universidad Eafit
Medellín, Colombia
scartagena@eafit.edu.co

Paulina Pérez Garcés
Universidad Eafit
Medellín, Colombia
pperezg@eafit.edu.co

3) Practice for final project defense presentation

3.1 For this exercise a tree where a branch can be the father for more than two others was needed. Ones first idea might be something similar to a B-Tree, but the data needed to be sorted with non-numeric values so it was no use. Nevertheless, we created something similar, where the “sons” of a value are stored in an array list; therefore, searching for something in our tree has a complexity O(n).

3.2 Yes, is it plausible to implement as a more efficient way, but if we analyze carefully the complexity will be O(log(n)) in most of the cases, but anyway is not possible to keep this complexity due to always in the worst case it will be O(n).

3.3 At the beginning we just started to build the BST(Binary Search Tree) based on the given array of ints for the input example of the exercise, this makes easier the exercise and assures us to keep the tree with the same structure, and this happens just because of the appearance of the first element of the preorder as the root of the tree, we used this fact as a starting point and also following the Binary Search Tree logic, placing minor data on the left side, and the major data on the right side, this using the help of an insert method created.Then, when we have already built the tree, just missed to implement a post-order method which allows us to reached to the minor node value and it goes back and describing a behavior as follows: left, right, root and continues successively, at the end it prints the root. In order to see the functioning of the algorithm previously described, you need to make a call to method for traveling the tree by the left side until the null is made, when this occurs you need to make a call to the method for walking the tree from back to ahead by the right side and that is it.

3.4 The complexity of a program that gets you the post-order sorting, taking into account that creating and filling the tree has a complexity of O(n) just like the method that prints the desired order, is O(n) thanks to the addition rule.

4) Practice for midterms

4.1 a*)* altura(raiz.izq)+1.

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

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

4.2 c.
4.3 a) suma == a.dato
b) a.data
c) a.izq, suma - a.data
d) a.der, suma – a.data
4.4
4.4.1 c. 2T (n / 2) + C
4.4.2 O(n)
4.4.3 d.
4.4.4 a
4.7.
4.7 1. a. 0,2,1,7,5,10,13,11,9,4
4.7.2 - 2
4.8 b.
4.9 a.
4.10 No
4.11
4.11.1 c
4.11.2 c
4.11.3 No
4.12
4.12.1 1,2,3,4,5,6,7,8,9,10
4.12.2 a.
4.12.3 O(n)
4.13
4.13.1 suma[raiz.id]
4.13.2 T(n - 1) + C, que es O(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