ESTRUCTURA DE DATOS 1
Código ST0245

Laboratory practice No. 4: Hash tables & trees

María Paulina Ocampo Duque
Universidad Eafit
Medellín, Colombia
mpocampod@eafit.edu.co

María José Gutiérrez Estrada
Universidad Eafit
Medellín, Colombia
mjgutierre@eafit.edu.co

3) Practice for final project defense presentation

3.1 The used structure in this algorithm was the hash tables,this works to test collisions and to calculate the bees in a specific position. When the bee is located in the position, this data will be saved in a linked list so that only one bee can exist in this location, then the bees will be test to see if there´s risk of collision, all of this using a recursive function

3.3 The algorithm recursively traverses each node in the tree to the left and right and when it finishes it prints the data for the current node.
The algorithm requested can find the pre-order tour. We create the node class (we need this to create the binary tree), then we create the methods in which we can:
insert a value, find if a specific value exists, print the InOrder, print the PreOrder, etc.
How does the algorithm work?
First,It prints the value of the current node then it moves to the left node and then to the right node

3.4 Complexity 2.1

The complexity in the worst case would be O(Log(n))

```
public void posOrder(Node node) {
    if(node != null) {                  //c1
        posOrder(node.left);            //c2*(n/2)
        posOrder(node.right);           //c3*(n/2)
        System.out.println(node.data); //c4
    }
}
```

T(n)=c1+c2*(n/2)+c3*(n/2)*c4
T(n)=O(c1+c2*(n/2)+c3*(n/2)*c4)
T(n)=O(2*(n/2))
T(n)=O(Log (n))

3.5 Variable 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

In the algorithm 2.1 the only variable is n and it is the number of elements of the binary tree since to print the desired order we must pass through the data

4) Practice for midterms

4.1.1 b
4.1.2 d
4.2 c
4.3 a. return false
b. return suma==0;
c. return sumaElcamino (a.getLeft(),suma -a.getValue())
d. sumaElcamino(a.getLeft (),suma);
4.4 a. b
b. b
c. d
4.5 a. toInsert == null
b. toInsert > p
4.9 a
4.11.1 b
4.11.2 a
4.11.3 b
4.13.1 raiz.id
4.13.2 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