

## Laboratory practice No. 4: Hash Tables and Trees

**Samuel Ceballos Posada**  
Universidad Eafit  
Medellín, Colombia  
sceballosp@eafit.edu.co

**Pedro Botero Aristizábal**  
Universidad Eafit  
Medellín, Colombia  
pboteroa@eafit.edu.co

### 3) Practice for final project defense presentation

- 3.1** We implemented a BTS (binary search tree). The idea behind this data structure is to have an efficient way to store, search and retrieve data. The Search time is  $O(\log n)$ .
- 3.2** If the tree were to have a balanced structure, the time for Search and Insertion would be  $O(\log n)$ . While traveling the tree in order the time would be  $O(n)$ .
- 3.3** First, the program starts to complete the tree inserting the integers that are placed on the data input. The first integer that gets inserted is used as the root and the rest is put on a specific position of the tree based on if the number is greater or smaller than the actual root based on a recursive model, following the definition of a Binary Search Tree. After the tree is complete, we now have to start printing it in a post-order way. This means that we have to print first the recursive left of the root, then the recursive right of the root and finally, the root itself. This way, we can see on a more efficient way the positions in which is number is located in the array and in that way, offer a simpler reading of the tree for the user himself.
- 3.4** Taking into account the fact that creating and organizing a tree has a complexity of  $O(n)$ , the method used to print that same tree in post-order also has a complexity of  $O(n)$ .

### 4) Practice for midterms

**4.1** b) The chains that start with the same letter collision and d)  $O(1)$ .

**4.2** c) 3

**4.3** a) `suma==a.data;`

b) `a.data`

c) `a.left, suma - a.data`

d) `a.right, suma - a.data`

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

**4.4.2** a)  $O(n)$

**4.4.3** d)

**4.4.4** 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

**ESTRUCTURA DE DATOS 1**  
**Código ST0245**

**4.7.1** a) 0, 2, 1, 7, 5, 10, 13, 11, 9, 4

**4.7.2** b) 2

**4.7.3** d)  $O(n)$

**4.8** b) 2

**4.9** a) 5, 3, 6, 1, 7, 4, 8, 0, 2

**4.10** b) No

**4.11.1** b) 2, 3, 4, 0, 5, 7, 6

**4.11.2** a) 5

**4.11.3** b) No

**4.12.1** i)  $A = 1, B = 2, C = 3, D = 4, E = 5, F = 6, G = 7, H = 8, I = 9, J = 10$

**4.12.2** a) G, D, B, A, C, E, F, I, H, J

**4.12.3** b)  $O(\log(n))$

**4.13.1** suma[raíz.id]

**4.13.2** a)  $T(n) = T(n-1) + C$ , 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

