**Laboratory n.4  
Hash tables and trees**

|  |  |
| --- | --- |
| **Miguel Ángel Martinez Florez**  Universidad Eafit  Medellín, Colombia  mamartinef@eafit.edu.co | **Pablo Maya Villegas**  Universidad Eafit  Medellín, Colombia  pmayav@eafit.edu.co |

**3) Simulacro de preguntas de sustentación de Proyectos**

**3.1** We used an octree data structure because it’s highly efficient at separating thousands of objects in a 3D space, and works perfectly with the bees coordinates. Its complexity in the worst case scenario its O(8^n) with n being the depth of the octree, in the worst case every single tree has enough bees to be able to divide again so n = Log8 m, with m being the amount of data, or bees inserted.

However, this data structure has a big problem regarding the bees, if two bees in close proximity get separated into different quadrants, it’s assumed that they aren’t close, which is false.

**3.2**

**3.3**

**3.4** In the binary tree point 2.1, creating the tree has a complexity of O(log n) and printing the post order string has a complexity of O(n) with n being the amount of data.

***4) Simulacro de Parcial***

4.1: B

4.1.2: D

4.2: C

4.3:

Linea 3: false

Linea 5: 0

Linea 7: sumaElcamino(a.der,suma-a.dato)

Linea 8: sumaElcamino(a.izq,suma-a.dato)

4.4.1: C

4.4.2: A

4.4.3: D

4.4.4: A

4.5:

Línea 4: p.dato == tolnsert

Línea 6: p.dato > tolnsert

4.6.1: D

4.6.2: return 0

4.6.3: == 0

4.7.1: A

4.7.2: B

4.9.1: A

4.11.1: B

4.11.2: B

4.11.3: A

4.12.1: A

4.12.2: A

4.12.3: A

4.13.1: suma[raíz.id]

4,13,2: D

***5) Lectura recomendada (opcional)***

Binary Trees

Binary trees are one of the fundamental data storage structures used in

programming.

Ways to Traverse a Binary Tree

How is a Slow Insert in an ordered array?

How do you use a Slow Insert in an ordered array?

Binary trees are non-linear data structures, they are considered as hierarchical structures

It consists of an array in which all the elements are arranged in order; that is, an ordered matrix.

an array in which all elements are ordered; that is, an ordered matrix, it is quick to find such a matrix for a particular value, using a binary search.

Why using binary trees has advantages?

and

as such, their way of traversing them differs substantially compared to linked lists which are linear type data structures.

because it combines the advantages of two other structures: an ordered array and a linked list.

It is important to clarify that it is faster to search an array of this type for a particular value, using a binary search.

An interesting and quick method is to iterate through an ordered array, visiting each object in ordered order.

fundamentally you can search a tree quickly, since

you can an ordered array, and you can also insert and delete items quickly, as you can with a linked list.

**6)** **Trabajo en Equipo y Progreso Gradual (Opcional)**

***6.1*** *Actas de reunión*

***6.2*** *El reporte de cambios en el código*

***6.3*** *El reporte de cambios del informe de laboratorio*