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 $n = \text{Log}_8 m$, 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 Genealogic trees aren't optimal for this problem, an alternative solution to this problem that has a complexity of n = n – 1, in which the bees are in a stack, and compares the top bee to the rest to check if they are in proximity to any, then pops the bee and repeats the process. This class it's called StackBees in the repository https://github.com/pmayavi/ST0245-002/blob/master/laboratorios/lab04/codigo/StackBees.java
3.3 The binary tree from 2.1, receives a string from which it separates the values and creates the tree root as the first value, then it enters the rest of the values to form the tree, the posOrderString method is called and it uses recursivity to travel the tree and print the string tree in pos order.
3.4 In the binary tree point 2.1, creating the tree has a complexity of O(n log n) and printing the post order string has a complexity of O(n) with n being the amount of data.

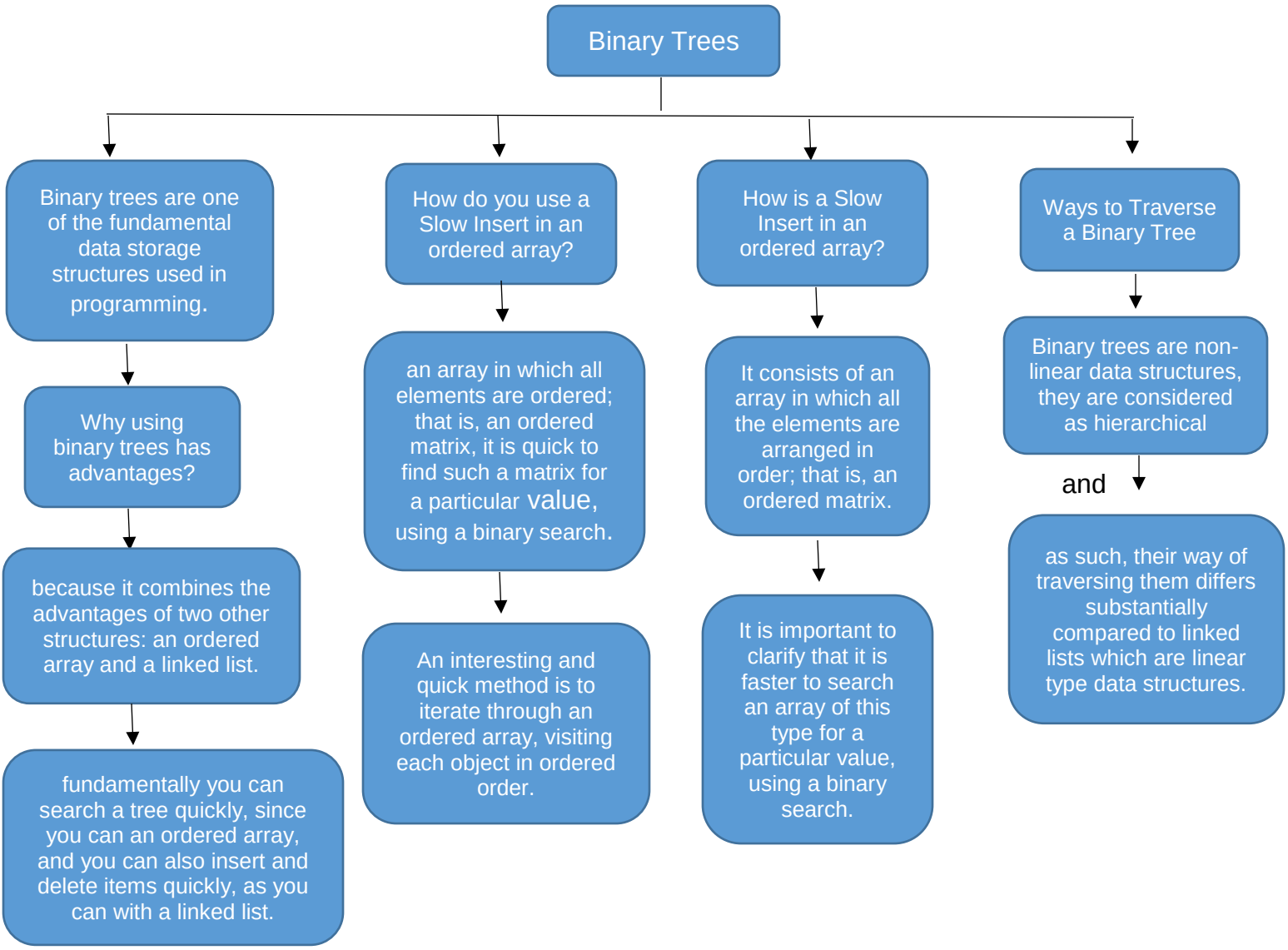
PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

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

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

5) Lectura recomendada (opcional)



PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473