

Laboratorio Nro. 4: Arboles

Santiago Pulgarin VasquezUniversidad Eafit
Medellín, Colombia
spulgarinv@eafit.edu.co**Juan Camilo Guerrero Alarcon**Universidad Eafit
Medellín, Colombia
jcguerrera@eafit.edu.co**3) Simulacro de preguntas de sustentación de Proyectos**

1. Concretamente si se puede implemetar un mecanismo o mejor dicho una estructura de datos mas eficiente que la que veniamos usando denominada Binary tree, en casos en los que no se requiere un mayor esfuerzo trabajo muy bien y sin problemas, trabajando incluso sobre tiempos excelente espero en el peor de los casos o con datos mas complejos no nos ayuda demasiado ya que actuaría un poco mas lento que una tabla de hash. Sin embargo, para que nuestro arbol genealogico funcione aun mas rápido existe una alternativa la cual es el AVL tree, uno de los primeros arboles implementados, y que ofrece una velocidad y tiempos mas optimos a grandes cantidades de datos, sin ningun problema podemos realizar nuestro arbol genealogico en tiempos de $O(n\log n)$, y se ajusta a nuestro ejercicio ya que principalmente trabaja con operaciones basicas.
2. El funcionamiento de este programa se basa en que inicialmente se introduce un arbol binario de busqueda en Pre-Orden y debe imprimirla en Pos-Orden, ¿Como se logra esto? Primero que todo hay que almacenar en un arbol binario de busqueda (Pre-Orden) y luego mediante el metodo ImprimirEnPosOrden() se imprime en Pos-Orden sabiendo que el primer dato introducido (root) en el Pre-Orden sera el ultimo en el Pos-Orden.
3. $O(n)$
4. En este caso la variable que existe es Node el cual indica la cantidad de nodos que tiene el arbol y dependiendo de dicha cantidad el arbol tiene " n " complejidad.

4) Simulacro de Parcial

1. altura(raiz.izq)
altura(raiz.der)
2. C
3. false
a.dato
a.izq, suma - a.dato
a.der, suma - a.dato
4. C
A
D
A
5. p.dato == toInsert

toInsert > p.dato

- 6.** D
Return 0
== 0
- 7.** 1
2
- 8.** B
- 9.** A
- 10.** B