Tecnológico de Monterrey
Escuela de Ingeniería y Ciencias

**Campus Sonora Norte**

# Act 3.1 - Operaciones avanzadas en un BST

Curso:

Programación de estructuras de datos y algoritmos fundamentales

(TC1031)

Estudiante:

Daniel Alfredo Barreras Meraz  A01254805

Docente:

Baldomero Olvera Villanueva

Fecha de entrega:

11 de octubre de 2023

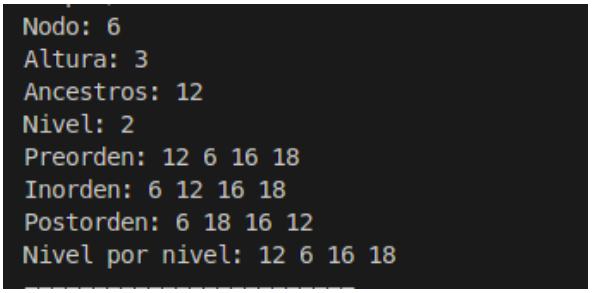
Código fuente

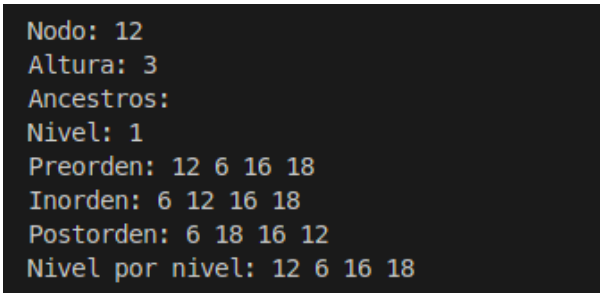
- Adjunto en el archivo zip.

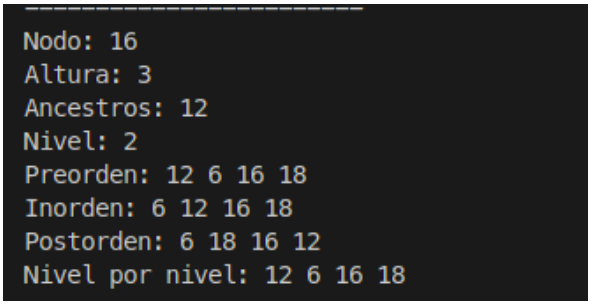
**Casos de prueba**

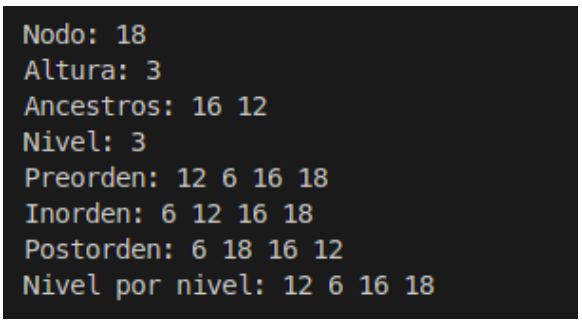
| **Caso de prueba** | **Entrada** | **visit** | **height** | **ancestors** | **whatlevelamI** |
|---|---|---|---|---|---|
| 1 | 6 | **Preorden:**<br>12 6 16 18<br>**Inorden:**<br>6 12 16 18<br>**Postorden:**<br>6 18 16 12<br>**Nivel por nivel:**<br>12 6 16 18 | 3 | 12 | 2 |
| 2 | 12 | | 3 | 0 | 1 |
| 3 | 16 | | 3 | 12 | 2 |
| 4 | 18 | | 3 | 16, 12 | 3 |

**Comprobación de casos de prueba**




**Complejidad temporal:**

**visit: O(1)** - La función visit se utiliza para acceder a un nodo del árbol y realizar una operación en él, como imprimir su valor. La complejidad es constante porque solo se necesita acceder a un nodo y realizar una operación simple en él.

**Preorden, Inorden, Postorden: O(n)** - Estos recorridos son algoritmos para visitar todos los nodos del árbol. En el peor caso, tendrás que visitar todos los nodos una vez, lo que lleva a una complejidad de O(n).

**Level by level: O(n) -** Este recorrido por niveles implica visitar todos los nodos del árbol siguiendo el nivel, y, en el peor caso, tendrás que visitar todos los nodos una vez, lo que lleva a una complejidad de O(n).

**height: O(n) -** Calcular la altura del árbol en el peor caso implica visitar todos los nodos lo cual le da una complejidad de O(n),  especialmente si el árbol está desequilibrado y se asemeja a una lista enlazada.

**ancestors: O(n) -** Para encontrar los ancestros de un nodo, en el peor caso, tendrás que recorrer el árbol desde la raíz hasta el nodo en cuestión, lo que implica visitar todos los nodos y, por lo tanto, una complejidad de O(n).

**whatlevelamI: O(n) -** En el peor caso, para encontrar en qué nivel se encuentra un nodo, tendrás que recorrer el árbol desde la raíz hasta el nodo en cuestión, lo que implica visitar todos los nodos y, en consecuencia, una complejidad de O(n).