

Laboratorio 4: General Trees/Queue (GTree/Queue)

Nicolás Murúa - Fernando Santolaya - Gilberto Gutiérrez

25 de Abril

1 Introducción

Este laboratorio de **EDD** se enfoca en implementar una navegación iterativa con Queue para un General Tree. La navegación se deberá hacer de forma iterativa, **evitando la recursividad**. Para profundizar la navegación y manipulación del GTree, se entrega un desafío opcional que le suma la eliminación de los nodos hoja. Se les entregará un proyecto con una colección de Estructuras de Datos que se han presentado durante el curso, por lo que este texto no profundizará en la funcionalidad de cada archivo. Los archivos relevantes son:

1. **GTreeBase.java**: Clase base con la estructura GTree y algunas utilidades básicas.
2. **TreeNode.java**: Clase con la estructura de Node para utilizar en el GTree.
3. **GTree.java**: Clase en la que deberá implementar la función de navegación y la función de eliminación.
4. **Evaluator.java**: Evaluador automatizado típico en nuestros laboratorios. El algoritmo a implementar es **Insert**. Este se profundizará en la sección 3.

2 Objetivos

1. Comprender de manera práctica las características de un General Tree.
2. Profundizar el uso de Queue como una herramienta para algoritmos de navegación.
3. Diseñar un algoritmo de navegación LevelOrder basado en Queue para nuestra clase GTree.
4. Implementar el algoritmo de navegación diseñado en la clase GTree.
5. **Opcional**: Profundizar el trabajo de navegación con un algoritmo que incluya la eliminación de las hojas.

3 Instrucciones

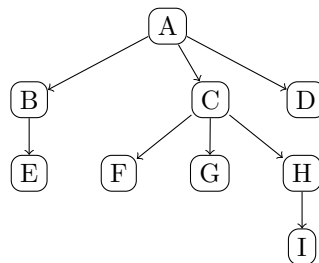
Este laboratorio no es evaluado y consta de 2 algoritmos que deberán implementar. La navegación **LevelOrder** es obligatoria, mientras que la función **deleteLeafs** es un trabajo adicional para perfeccionar el trabajo con General Trees y Queues. A continuación se presentan los detalles:

3.1 LevelOrder

Esta función se debe implementar de forma iterativa utilizando una Queue para realizar un recorrido Level Order (por nivel) sobre un General Tree. En este tipo de recorrido el algoritmo deberá:

1. Visitar la raíz del árbol.
2. Recorrer todos los hijos de la raíz (Todo el nivel).
3. Repetir con los hijo de esos nodos siguiendo el mismo order.

Por ejemplo, para un GTree con la siguiente estructura:



El algoritmo deberá retornar una Queue que contenga los nodos ordenados por nivel como se presenta a continuación:

Queue output = GTree.deleteLeafs()
output == [A] → [B] → [C] → [D] → [E] → [F] → [G] → [H] → [I]

3.2 DeleteLeafs

Esta función se debe implementar con un recorrido LevelOrder. El desafío está en que este algoritmo debe eliminar los nodos hoja (sin hijos) manteniendo los punteros del árbol correctamente. Los nodos eliminados deberán ser retornados en forma de una Queue (de la misma forma que se utilizó en el algoritmo de navegación). Si al momento de eliminar un nodo hoja su padre se convierte en hoja no se deberá eliminar pues al momento de invocar la función no era un nodo hoja.

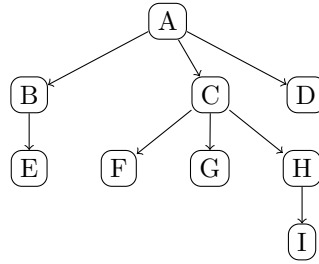


Figure 1: Estructura del GTree antes de eliminar los nodos hoja

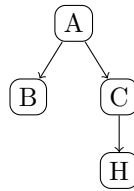


Figure 2: Estructura del GTree después de eliminar los nodos hoja

Por ejemplo, para un GTree con la siguiente estructura presentada en 1, el GTree resultante será el de 2.

Además se deberá retornar una Queue con los nodos eliminados que conserve el orden en el que fueron eliminados (LevelOrder) como se presenta en 3.

$$Queue\ output = GTree.deleteLeafs()$$

$$output == [D] \rightarrow [E] \rightarrow [F] \rightarrow [G] \rightarrow [I]$$

Figure 3: Queue retornanda por la funcion deleteLeafs() con los nodos hojas eliminados.

Deberá completar el código para la función **LevelOrder** de modo que pase la todos los casos de prueba extra de la clase **Evaluator**.

4 Consideraciones

La actividad de no es evaluada pero se exige que entregue el archivo **GTree.java** con la implementacion realizada. Los modelos de lenguaje (chatGPT, claude, deepseek, etc) no están prohibidos, pero se recomienda que su uso se limite a orientar el pensamiento en lugar de copiar y pegar su respuesta. En caso de utilizarlo, el/la estudiante deberá incluir comentarios en el código señalando que partes fueron apoyadas por IA y como construyo su solución sobre esto.