

Laboratorio N°4

Estructura de Datos y Algoritmos

Fecha de inicio 29 de mayo 2023 desde las 23:59h

Fecha de entrega 05 de junio 2023 hasta las 23:59h

Enunciado

En la empresa quieren reestructurar la plataforma para poder incorporar un buscador de vídeos basado en el nombre de los canales. Para esto se agrupan los vídeos por canal con base en el campo "ChannelTitle". Posteriormente, las listas de cada canal se organizan en un árbol AVL. Usted recibe una implementación de este tipo de árbol que dejó un trabajador que renunció, por lo que deberá depurar e implementar los métodos indicados. Adicionalmente, deberá incluir los métodos incorporados en el cliente y probar el funcionamiento con los cambios integrados. Este, en el peor de los casos, debe realizar la búsqueda de un canal del árbol en complejidad $O(\log n)$.

Descargue el *dataset* y los códigos de los siguientes links:

Dataset: <https://www.kaggle.com/datasets/danielaamoren/youtubevideots>

Códigos: <https://drive.google.com/drive/folders/1BcBvhKULwY1YvcA4a3n4TcuTdGalsyHM?usp=sharing>

1 Documentación Clase AVLtree

- **AVL(String file):** constructor de AVL, este realiza la separación de los campos del csv y luego lo agrega a una HashTable de la librería de *java.util.Hashtable*. El *hash* tiene como llave el nombre del canal y guarda una lista enlazada de los vídeos correspondientes al canal.
- **insert(String key, LinkedList videos):** inserta un canal en el AVL.
- **balance(Node node):** el método se encarga de actualizar las alturas desde el subárbol entregado. Luego, revisa si el nodo que recibe de entrada está desbalanceado, en caso de que este desbalanceado realiza las rotaciones respectivas a cada caso (LL, LR, RR, RL).
- **rotateLeft(Node n1):** realiza la rotación hacia la izquierda considerando n_1 , n_2 y n_3 , tal y como se ve en la figura 1.
- **rotateRight(Node n1):** realiza la rotación hacia la derecha considerando n_1 , n_2 y n_3 , tal y como se ve en la figura 1.
- **find(String key):** recibe como llave el título del canal y retorna la lista enlazada de vídeos correspondiente al mismo. En caso de no encontrar el canal retorna *null*.
- **delete(String key):** recibe como llave el título del canal y lo elimina del árbol. Este método no tiene retorno, por lo que en el cliente se debe verificar primero que el nodo se encuentre con el método *find* y luego efectuar la eliminación.

2 Requerimientos

Ignore las clases *LinkedList*, *MaxHeap*, *Minheap* y *Video* estas se encuentran para otras funcionalidades ya implementadas del menú, puede modificarlas solo en caso de requerirlo para el nuevo requisito de la plataforma.

Específicamente, en este laboratorio deberá realizar:

- Depuración del constructor y de los métodos de **insert**, **balance**, **rotateLeft** y **rotateRight** del árbol AVL. Comente sobre la depuración en el informe.
- Implementación de los métodos **delete** y **find** junto con un menú de búsqueda que permita al cliente buscar el nombre del canal. Comente sobre la implementación de los métodos en el informe.

3 Condiciones de entrega

- La tarea se desarrolla en **parejas**. Los códigos serán sometidos a comprobación automática de plagio. En caso de copia entre estudiantes se sancionará con la nota mínima.
- El código debe estar escrito en Java y **se debe poder compilar y ejecutar sin cambios**. De no cumplir esto será evaluado de forma inmediata con la nota mínima.
- El informe debe ser entregado en formato **PDF**. Realizar el informe es requerimiento, si el informe está vacío será calificada con la nota mínima.
- **Se deben entregar todos los programas y el informe en un archivo .zip** con el nombre del grupo vía canvas. Cualquier entrega atrasada no se considerará y será calificada con la nota mínima de inmediato.

4 Ejemplos de salidas

4.1 Rotaciones

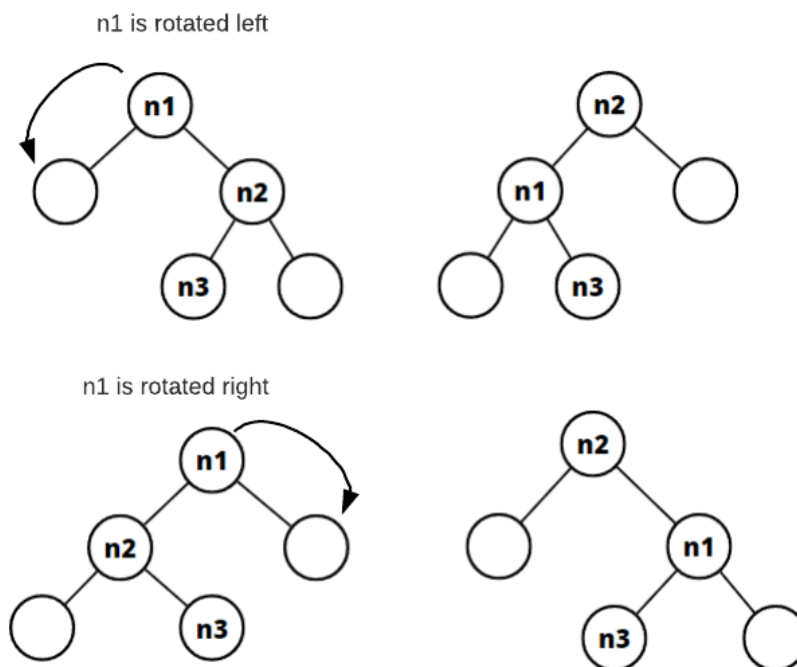


Figure 1: Ejemplos de rotaciones en un árbol AVL.

4.2 Inserción y eliminación

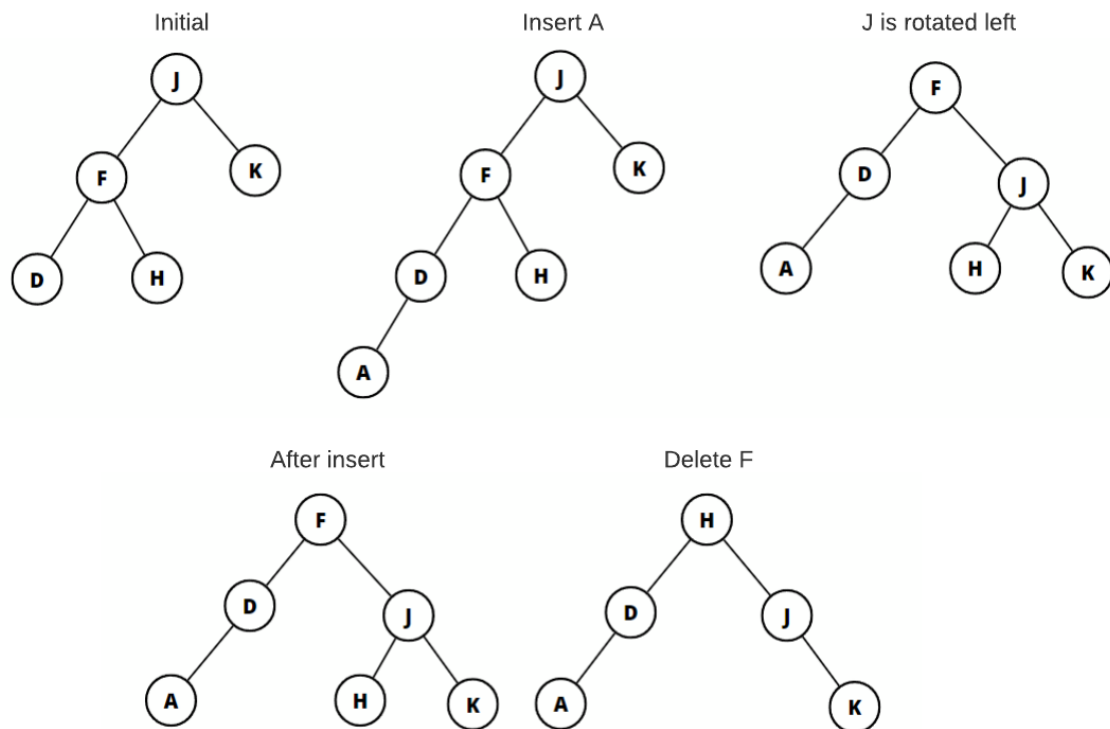


Figure 2: Inserción y eliminación en el árbol AVL de ejemplo en AVLTree.java.

4.3 Salida correcta de datos de prueba en AVLTree.java.

Datos de prueba:

Árbol inicial:

Channel Title: J| Height: 2| root: video 1

Channel Title: F| Height: 1| root: video 9

Channel Title: D| Height: 0| root: video 4

Channel Title: H| Height: 0| root: video 7

Channel Title: K| Height: 0| root: video 6

Luego de insertar nodo A:

Channel Title: F| Height: 2| root: video 9

Channel Title: D| Height: 1| root: video 4

Channel Title: A| Height: 0| root: video 10

Channel Title: J| Height: 1| root: video 1

Channel Title: H| Height: 0| root: video 7

Channel Title: K| Height: 0| root: video 6

Luego de eliminar nodo F:

Channel Title: H| Height: 2| root: video 9

Channel Title: D| Height: 1| root: video 4

Channel Title: A| Height: 0| root: video 10

Channel Title: J| Height: 1| root: video 1

Channel Title: K| Height: 0| root: video 6

Resultado de búsqueda: A primer vídeo: video 10

Resultado de búsqueda: null