

Facultad de Ciencias, UNAM

Práctica 4: Árboles

Alejandro Hernández Mora

Jerónimo Almeida Rodríguez

Adriana Sánchez del Moral

Kethrim Guadalupe Trad Mateos

Fecha de entrega: Por definir

Descripción

La entrega de esta práctica es individual.

Lee con atención las instrucciones y resuelve.

En esta práctica desarrollarán la implementación de un Árbol Binario en su versión de árbol de búsqueda y de árbol balanceado AVL.

Antes de iniciar la práctica, ejecuta los comandos:

- `$ cd src`
- `$ javadoc -d docs ArbolAVL.java`

para generar la documentación de las clases de la práctica.

Su trabajo consiste en implementar los métodos de las clases `ArbolBinario`, `ArbolBinarioBusqueda` y `ArbolAVL` y probar que su comportamiento sea el adecuado.

De la clase `ArbolBinario` tienen que:

- Implementar los nodos del árbol.
- Implementar los constructores que reciben argumentos
- Implementar los métodos `altura`, `getTamano`, `raiz`, `esVacia`, `vaciar` y `equals`.
- Implementar los algoritmos `preOrden`, `postOrden` e `inorden` junto con sus métodos auxiliares.
- Cómo punto extra, implementar el `toString`.

La clase `ArbolBinarioBusqueda` extiende `ArbolBinario` y tiene:

- El iterador a implementar.
- Los métodos constructores de la clase que reciben argumentos.
- Los métodos `agregarNodo`, `agregar`, `eliminarNodo`, `eliminar` y `buscaNodo`.
- El método `maximoEnSubarbolIzquierdo` que busca el elemento más grande del subárbol izquierdo.
- Los métodos `giraDerecha` y `giraIzquierda` que giran el árbol sobre el nodo que reciben.
- El método `iterator`.

La clase `ArbolAVL` extiende `ArbolBinarioBusqueda` y tiene:

- Su propio nodo que extiende el nodo de arbol binario.
- El iterador a implementar.
- Los métodos constructores de la clase que reciben argumentos.
- Los métodos `actualizaAltura`, `getAltura`, `maximoEnSubarbol`, `eliminar` y `nodoAVL`.
- El método `rebalancea` que se encarga de rebalancear el árbol según su altura.

Condiciones de entrega

1. Descargar el repositorio que les fue asignado desde la siguiente liga:
2. Programar los métodos especificados en la clase Lista.java y verificar que estos funcionen adecuadamente.
3. Mantener una estructura limpia de su práctica, con el código en la carpeta **src** y los documentos en la carpeta **src/docs**
4. Su código debe seguir buenas prácticas de programación, ser legible y estar bien indentado.
5. Subir su código a github.
6. Subir a github un archivo readme.md con su nombre completo, descripción del trabajo y pruebas de ejecución.
7. Entregar tarea en google Classroom (no es necesario subir nada)