Correccion - Gonzalez Calderon, Julian - TP ABB:

Segundo cuatrimestre 2021

Esta es la correccion de la primer entrega del TP de Arbol Binario de Busqueda, en ella vas a encontrar comentarios generales, sobre tu codigo, sobre el informe que entregaste y finalmente sugerencias para futuras entregas. Al final del documento, vas a encontrar tu nota. En este momento se cambiaron los correctores, pero ya hay bastante camino recorrido en cuanto a los TPs, la correccion y el aprendizaje, por lo que algunas correcciones que se arrastren de antes y no eran tan graves, empiezan a ser mas notorias; acordate que la idea es ir mejorando en base a las correcciones hechas anteriormente. Si surgen cosas nuevas, es normal debido al aumento de complejidad en los temas, asi que no te preocupes si ves mas comentarios. Como siempre, te pedimos que le prestes atencion a la devolucion del TP. Me interesa que leas los comentarios independientemente de la nota final, que al final del dia, es solo un numero :)

En general:

- Se cumple la mayoria de buenas practicas de codigo.
- Problemas con extraccion de codigo en archivos (ver mas abajo).
- Codigo Prolijo y limpio, buen trabajo!
- Muy buenas pruebas.

README:

- Nota sobre ABB: en General, el estandar es que en la rama izquierda estan los MENORES, y en la derecha los MAYORES. Lo raro es que en el codigo respetaste esa logica, y en el readme lo pusiste al reves.
- Nodo Padre: El nodo padre no es la raiz del arbol principal, ojo con eso. Desde cualquier nodo, el nodo inmediatamente superior conectado, es el padre. Mientras que el nodo raiz es UNICO, y es el unico nodo en el 1er nivel del arbol.
- Acordate de mostrar las funciones que te parezcan mas importantes o que consideres bueno explicar, no es necesario hacerlo con todas.
- Muy buenos diagramas!

Código:

- No entiendo porque extrajiste la logica de las funciones del .c del ABB hacia otro archivo interno. Esa forma de separar el codigo solo lo alarga. Pensa que la logica del ABB tiene sentido que este en el mismo archivo abb.c. Fijate que las funciones auxiliares terminarian siendo las funciones con logica.
- Return en funciones void: Las funciones void no necesitan return de ningun tipo. Se pueden poner, pero solo complica la legibilidad del codigo. cuando tenes sentencias de if con returns, lo mejorque podes hacer es invertirlas, y poner dentro del if el codigo que se ejecuta en el caso de que se tenga que ejecutar, y si no entra en el if, la funcion termina y listo.
- La funcion de destruir el arbol puede usarse reutilizando el quitar nodo, siempre sacando la raiz hasta que no haya mas elementos. De esta manera reutilizas codigo y queda mucho mas claro.

- Devolver ERROR cuando no es el caso: La funcion de tamanio del abb devuelve una constante llamada ERROR en el caso de que no haya elementos, ojo con esto. No es un error, esta perfecto tener una constante de error, pero no porque valga cero, tenes que usarla para devolver cantidades, la cantidad 0 no es un error, es una cantidad valida, y si un dia cambias la constante ERROR porque ahora el codigo de error es 404 (por ejemplo), la funcion de tamanio pasaria a devolver algo invalido.

Pruebas:

- Ojo con los mensajes de las pruebas, en una pusiste "Tamaño de un arbol vacio devuelve NULL", lo cual es incorrecto, devuelve cero, pero la prueba devuelve cero. Cuidado porque al ejecutar la prueba, su afirmacion indica el correcto comportamiento.
- Fuera de eso, muy buen nivel de pruebas :)

Sugerencias:

- Revisa un poco la teoria de ABB, sobretodo para repasar algunas definiciones.
- Ojo cuando extraes a otro archivo funciones, o extraes funciones a auxiliares, hay un punto donde es util, y otro donde ya es codigo menos claro, fijate que siempre que lo hagas, aporte algo y tenga una razon para hacerse.

Nota: 1.75/2. Buen TP! :)