Actividad integradora 3. Árboles binarios

Integrantes:

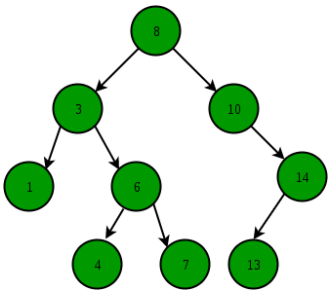
Armando Arredondo Valle - A01424709

30 Octubre de 2022

En lo que respecta a la complejidad, es necesario realizar un análisis a fondo para la parte de tener una buena optimización, así como un buen desempeño del código.

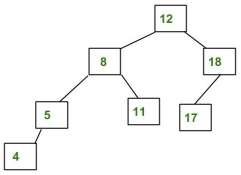
Binary Search Tree

En lo que respecta a los BST, es posible que el tiempo de la complejidad y la inserción de operaciones puede ser de O(h), donde la “h” habla sobre la altura del BST. Y en el peor de los casos, quizás tengamos que viajar desde la raíz hasta aquella hoja profunda que nos determina el fin. La altura puede convertirse en **n**, y la operación de inserción puede terminar siendo O(n).



AVL

La mayoría de las operaciones que respectan al BST, pueden tener tiempos de O(h), la ventaja de hacer uso de AVL’s es que la misma complejidad siempre será en todo momento O(log(n)), donde n dependerá del número de nodos dentro del mismo árbol.



Conclusiones:

El hecho de que exista el AVL, acorta significativamente los tiempos de ejecución, permitiendo tener una mejor optimización en lo que respecta al uso de BST.

Referencias:

- *AVL Tree | Set 1 (Insertion)*. (2022, August 22). GeeksforGeeks. Retrieved October 31, 2022, from https://www.geeksforgeeks.org/avl-tree-set-1-insertion/

- *Binary Search Tree*. (2022, October 25). GeeksforGeeks. Retrieved October 31, 2022, from https://www.geeksforgeeks.org/binary-search-tree-data-structure/