



# Tecnológico de Monterrey

**Programación de estructura de datos y algoritmos fundamentales TC1031 (Gpo 601)**

## **Actividad 3.4 – Reflexión personal**

**Profesor:**

Dr. Eduardo Arturo Rodríguez Tello

**Alumno:**

Nicolas Aguirre Villafañe A00832772

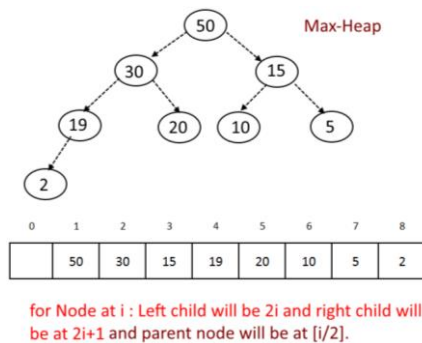
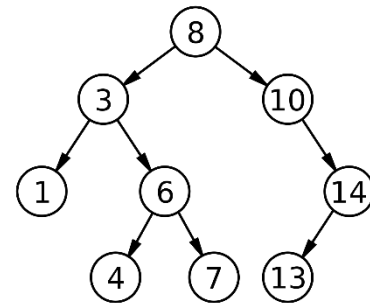
a 22 de mayo de 2022

## Investigación y reflexión:

En la ciencia de datos, tener una buena organización y distribución de los datos, es muy importante y de alta relevancia ya que de esto depende la eficiencia del programa y qué tan bueno puede ser. Aquí es donde entran los BST; estas estructuras de datos son unas de las mas importantes y simples para estas estructuras con miles de datos.

En un árbol de búsqueda binaria, el subárbol izquierdo tiene los valores mas bajos que el nodo y el subárbol derecho tiene valores mayores que el nodo. En este se pueden hacer 3 tipos de recorridos: post-order, pre-order y inorder. En esta búsqueda su eficiencia esta entre  $\lceil \log_2(N+1) \rceil$  y  $N$ .

Gracias a su estructura ordenada, los bst se pueden usar para muchas aplicaciones como indexing y multi-level indexing, aplicar algoritmos de búsqueda, mantener ordenada la data, etc.



Igualmente, el MaxHeap es un bst muy importante y fácil de usar y gracias a su fácil acceso se puede tener una alta eficiencia. Este nos ayuda a tener un fácil acceso al elemento más grande de una lista. Posteriormente, se puede eliminar este item y el maxheap se reorganiza automáticamente y se vuelve a poner el elemento mas grande siguiente. Un heap es básicamente un binary tree pero ordenado linealmente en un array. Gracias a su fácil acceso al elemento mas grande, su eficiencia es de  $O(1)$ .

Gracias a estos conocimientos de estructuras de datos, se puede hacer un análisis o un algoritmo para determinar si una red esta infectada al implementar un conteo de los registros. Ya que cada que alguien entra a una red, se crea un registro, se puede crear un contador de los registro, y si hay muchos registros con errores al entrar en un corto periodo de tiempo o que se vean sospechosos, se puede determinar si se está siendo atacado o no.

**Referencias:**

Árbol binario de búsqueda. (n.d.). Wikipedia. Retrieved May 23, 2022, from [https://es.wikipedia.org/wiki/%C3%81rbol\\_binario\\_de\\_b%C3%BAsqueda](https://es.wikipedia.org/wiki/%C3%81rbol_binario_de_b%C3%BAsqueda)

AVL Tree And Heap Data Structure In C++. (2022, May 4). Software Testing Help. Retrieved May 23, 2022, from <https://www.softwaretestinghelp.com/avl-trees-and-heap-data-structure-in-cpp/>

Skotar, M. (2019, September 29). Importance of Binary Search Trees | by Michael Skotar. Medium. Retrieved May 23, 2022, from <https://medium.com/@michaelskotar/importance-of-binary-search-trees-d354afc6e347>

When would I want to use a heap? (n.d.). Stack Overflow. Retrieved May 23, 2022, from <https://stackoverflow.com/questions/749199/when-would-i-want-to-use-a-heap>