

**Instituto Tecnológico y de Estudios Superiores de
Monterrey**



Programación de estructuras de datos y algoritmos fundamentales

**Tarea individual: Act-Integradora-3
Estructura de datos jerárquica**

Profesor: Dr. Eduardo Arturo Rodríguez Tello

Marlon Yahir Martínez Chacón - A01424875

16/05/2023

Importancia y eficiencia del uso de los BST

Los árboles binarios como varias estructuras de datos, nos permiten almacenar grandes cantidades de información de manera ordenada para poder manipularla como lo creamos más conveniente para resolver el problema que estamos enfrentando. En cuanto a los árboles, nos permiten dividir nuestros datos en varias secciones mediante los nodos hijos y padres, haciendo que podamos implementar una gran cantidad de funciones para manejar cada una de estas secciones, que pueden llegar a ser muy útil cuando se tiene mucha información porque existen una gran cantidad de formas de recorrer un árbol debido a la estructura que tiene. Existen varios tipos de árboles que cuenta con sus características propias, por lo que dependerá del tipo de problema que necesitamos resolver la elección del árbol binario.

Una de las características que hacen eficiente el uso de los diversos tipos de arboles binarios que existen es que al momento de crear su estructura ya se encuentra implementando de alguna manera algún algoritmo ya sea de ordenamiento o de búsqueda, por ejemplo en los árboles binarios de búsqueda su estructura está diseñada para poder realizar de manera más rápida la búsqueda binaria y en los max Heap que usamos una de sus principales características es que facilita el uso del heap sort, uno de los algoritmos más eficientes que existen por la complejidad de $O(n \log n)$.

- | | |
|-------------------|--|
| • Por intercambio | $O(n^2)$ |
| • Por inserción | $O(n^2)$ |
| • Por selección | $O(n^2)$ |
| • Merge Sort | $O(n \log n)$ |
| • Quick Sort | $O(n \log n)$ *su peor caso tiende a ser n^2 |
| • Heap Sort | $O(n \log n)$ |

Dentro de su implementación se encuentra la función heapify, que es muy usada para varios métodos del heap por ejemplo el pop, heap sort, entre otros, por lo que es más sencillo comprender como es que se realiza ese proceso al tener esa función en otros métodos, por lo que también es un punto a tomar en cuenta sobre la importancia de estas estructuras de datos.

En nuestro caso donde los servidores recibieron una cantidad masiva de hackeos, el tipo de árbol que estamos ocupando es el heap, en concreto el max Heap, ya que una de las necesidades que el cliente requiere es saber si una red esta infectada o no, por su naturaleza el maxHeap es lo ideal para resolver nuestro problema, esto es debido a que el nodo raíz del max Heap siempre va a contener el mayor valor de todo el heap, por lo que lo único que se necesita para obtener las ip a las que más se accedió durante el hackeo es estar sacando el nodo raíz del heap tantas veces como sean necesarias hasta tener la información necesaria para el cliente, y esto es posible porque el max heap se repara cada vez que se saca el nodo raíz para tener otra vez en la raíz el valor más grande.

En conclusión, para encontrar las ips con el mayor números de accesos y de esta forma saber cuáles redes están infectadas, el árbol ideal fue el maxHeap por todas sus características que permite manejar grandes cantidades de información, ordenarlas y centrarnos en el que tiene el mayor valor dentro de toda esa información, además de que los métodos que se implementaron y se usaron, son los que tienen la complejidad temporal menor a diferencia de otros que se implementaron con otras estructuras de datos, por lo que su manejo fue eficiente y rápido con toda esa información. Por lo que tomando esto como ejemplo me di cuenta que los árboles binarios tienen muchas posibilidades que se pueden explotar para obtener la información necesaria sin tener que sacrificar memoria por los procesos que se ejecutan y que cada problema que pueda surgir puede ser resuelto por algún tipo de árbol por los diversos tipos que existen.

Bibliografía

Árboles Binarios. (2014, 10 junio). Estructuras de datos.
<https://hhmosquera.wordpress.com/arbolesbinarios/>

LABORATORIO DE ESTRUCTURAS DE DATOS. (s. f.).
http://informatica.uv.es/iiguia/AED/oldwww/2002_03/Practicas/Practica8/pr_08_2003.html