

Actividad 3.4

Árbol Heap

Actividad Integral

TC1031.501:
Estructuras de datos y algoritmos fundamentales
Ing. Baldomero Olvera Villanueva

Tecnológico de Monterrey

Code and
Documentation
Repository

https://github.com/kochomati/tec_algoYDatos/tree/master/actividad_3_4

Arboles Heap

Un heap es un binary tree con propiedades especiales:

1. Todos los nodos deben estar en un orden específico
2. Su forma debe ser completa (todos los niveles menos el último)
3. El último nivel debe tener a todos sus nodos izquierdos llenos.

En un min heap, cada nodo padre contiene un valor igual o menor a los nodos hijos. En un max heap es lo opuesto.

Un heap puede tener valores duplicados y lo más importante es el principio de heap-order (condiciones padre-hijos max-min). El árbol tiene una altura mínima de $O(\log(n))$.

Esta estructura de datos tiene una complejidad de tiempo de $\log_2 n$. Es llenado de arriba para abajo y de izquierda a derecha.

Los hijos del nodo de índice i son $(2 * i + 1)$ y $(2 * i + 2)$

Operación	Complejidad (Worst)	Descripción
empty()	$O(1)$	Sólo verifica si hay datos o no.
insert()	$O(\log n)$	Mueve elemento insertado arriba p/ encontrar ubicación según valor.
getMax()	$O(1)$	Muestra valor de raíz.
removeMax()	$O(\log n)$	Elimina valor maximo y mínimo en raíz es reemplazado

Detección Redes Infectadas

Los accesos de manera muy frecuentes y a altas velocidades desde un mismo IP tienen alta chance de ser ataques maliciosos. La detección de ciertos patrones más complejos de accesos pueden ser detectados con machine learning o inteligencia artificial.

Para redes infectadas se deben quitar el malware de todos los dispositivos o identificar los servidores dummy para desactivarlos.