



Tecnológico de Monterrey

Luis Enrique Gutierrez

A00837481

Programación de Estructura de Datos y algoritmos Fundamentales (Gpo 850)

Profesor:

Dr. Eduardo Rodriguez Tello

Reflexión Actividad Integradora 3

Estructuras de Jerárquicas

Árboles Binarios y Heap Trees

Al indagar en el ámbito tecnológico la información puede ser guardada en un sinnúmero de tipos de algoritmos. En este caso estaremos hablando sobre estructuras de datos jerárquicas, los cuales se basan en tener una estructura parecida a la de un árbol dentro del cual cada nodo tendrá una raíz. Algunos ejemplos de este tipo de estructuras que han demostrado ser los más efectivos para resolver problemas de esta magnitud son los árboles de búsqueda binaria (BST) y árboles heap.

Un árbol BST es aquel cuya función principal es guardar información de manera ordenada. Podemos decir que el árbol se divide en dos columnas (lado izquierdo y lado derecho) donde los datos almacenados del lado izquierdo siempre serán menores que el valor de la raíz y sucediendo inversamente del lado derecho. Se pueden hacer un sinnúmero de operaciones como: búsqueda de elementos, inserción, eliminar. Todas estas operaciones nos ofrecen una complejidad $O(n)$ por lo que serán realizadas de manera rápida y eficaz. Lo anterior se puede ver afectado por algunos factores como lo puede ser repetición de valores, tiempo de ejecución al balancear el árbol cuando se inserta o se eliminan valores.

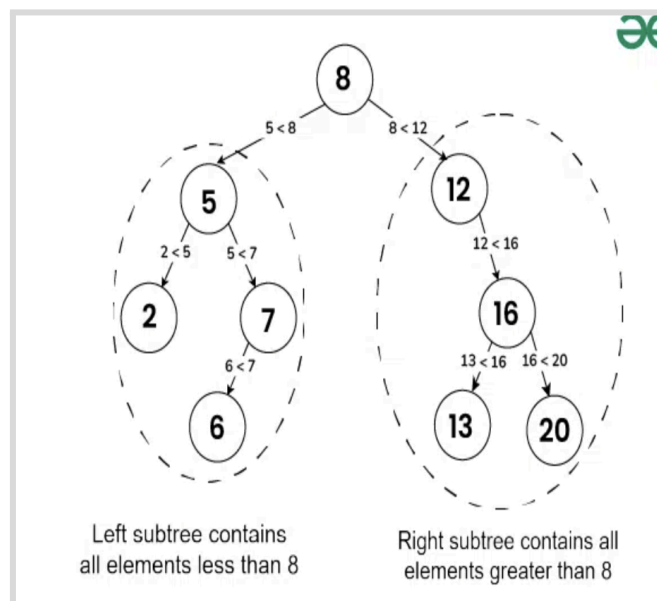


Imagen 1: Ejemplo de un BST

Obtenida de GeeksforGeeks

Por otro lado existe otra estructura de datos llamada árbol Heap la cual tiene como característica principal que el árbol sea llenado de forma ascendente o descendente en cuanto a los valores de sus nodos o el tipo de heap empleado. Podemos utilizar dos tipos de heap tree min heap dentro del cual la raíz tiene el valor más pequeño por lo que los valores de los

nodos (hijos) van a ir incrementado o max heap que tiene el valor de raíz más grande y los nodos (hijos) disminuyen cuando bajamos por el árbol. Como resultado, para todas las operaciones que podamos realizar un árbol heap tenemos una complejidad en general de $\log(n)$. Lo que lo hace un algoritmo muy eficaz y rápido para resolver problemas.

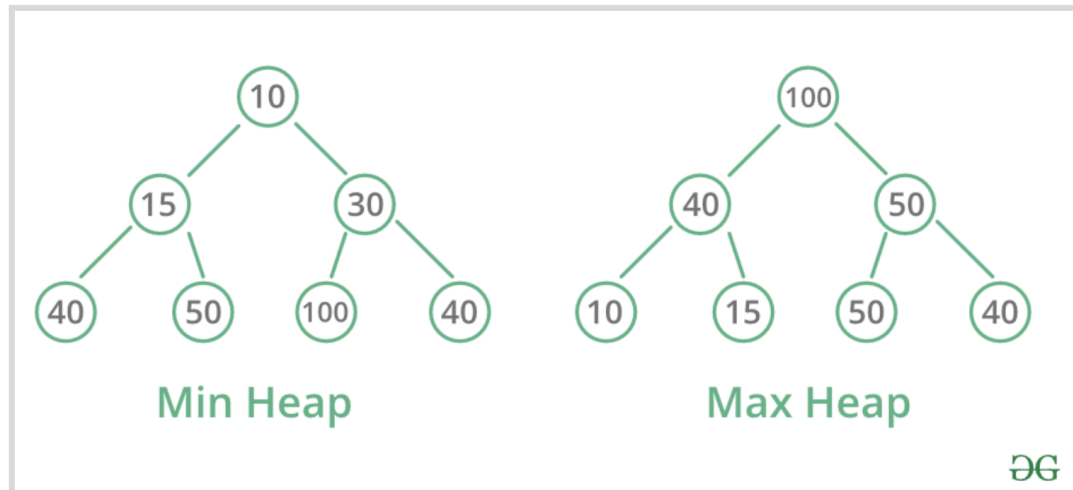


Imagen: 2 Diferencia entre Min Heap y Max Heap

Obtenida de Geek for Geeks

Cada tipo de estructura de datos tiene su respectiva complejidad computacional lo que afecta directamente al resultado. Es decir al tiempo que le tomara realizar las diferentes operaciones (push, getTop, pop). Por lo que si realizamos dichas operaciones en el árbol BST tendrán como salida una complejidad tipo $O(n)$ siendo no tan efectivo y eficaz como lo es utilizar un heap tree que tiene una complejidad de sus operaciones de $\log(n)$. También le podemos sumar otros factores sobre cual sería mejor de emplear, por ejemplo heap tree nos permite utilizar objetos teniendo un otro punto a favor sobre utilizar dicho método.

Se puede relacionar dicha información para saber cuando una red puede llegar a estar infectada por un agente externo. Típicamente se puede identificar con el desempeño que están teniendo los equipos, se vuelven más lentos o las aplicaciones no responden como deberían. Pero en mi opinión también se puede saber aplicando el concepto del proyecto al identificar una frecuencia de direcciones IPs que se repiten, ya que probablemente estén tratando de hacer un daño al sistema. Sabiendo esto podemos tomar medidas preventivas ante cualquier eventualidad.

```
ips_con_mayor_acceso.txt
1  10 IPs con mas accesos:
2  Elemento 1: 10.15.187.246, 38
3  Elemento 2: 10.15.183.241, 37
4  Elemento 3: 10.15.176.241, 38
5  Elemento 4: 10.15.176.230, 37
6  Elemento 5: 10.15.180.251, 35
7  Elemento 6: 10.15.185.220, 33
8  Elemento 7: 10.15.177.224, 37
9  Elemento 8: 10.15.182.223, 34
10 Elemento 9: 10.15.173.245, 35
11 Elemento 10: 10.15.171.225, 34
12
```

Imagen 3: IPs con mayor accesos

Durante el desarrollo de esta actividad se identificó cuál de las dos estructuras de datos más comunes de este ámbito. Aunque los BST han demostrado ser eficientes, en tareas como inserción, eliminación y búsqueda esto se puede llegar a ver limitado por tener una complejidad de $O(n)$ que no es la más rápida ni la más eficiente. Por otro lado, los heap trees ofrecen un resultado de tiempo de $\log(n)$ por lo que lo hacen más eficiente y rápido. Si a esta estructura le sumamos podemos incluir objetos se acoplara mejor a estos tipos de problemáticas. Esto lo pusimos en práctica al identificar cuales son las diez IPs que más intentaron acceder al sistema y se asocia al tema de prevenir riesgos de ataques. En conclusión heap tree demostró ser una estructura de datos sólida, no solo respaldada por su complejidad computacional sino también por poder ser utilizada en diferentes contextos para manejar todo tipo de información.

Referencias

GeeksforGeeks. (2020, June 26). Complexity analysis of various operations of Binary Min Heap.

<https://www.geeksforgeeks.org/complexity-analysis-of-various-operations-of-binary-min-heap/>

GeeksforGeeks. (2022, December 21). Complexity of different operations in binary tree, binary search tree and AVL Tree.

<https://www.geeksforgeeks.org/complexity-different-operations-binary-tree-binary-search-tree-avl-tree/>

GeeksforGeeks. (2024a, March 30). Heap Data Structure.

<https://www.geeksforgeeks.org/heap-data-structure/#what-is-heap-data-structure>

GeeksforGeeks. (2024, April 19). Introduction to binary search tree - data structure and algorithm tutorials.

<https://www.geeksforgeeks.org/introduction-to-binary-search-tree-data-structure-and-algorithm-tutorials/>

Inspired eLearning. (2020, March 17). How to tell if your router is infected & what to do about it.

<https://inspiredelearning.com/blog/router-is-infected-what-to-do/#:~:text=Common%20Signs%20Your%20Router%20May%20Be%20Infected%20By%20Hackers&text=Your%20computer%20is%20running%20more,up%20windows%20on%20your%20screen.>

Qué es un modelo de base de datos. Lucidchart. (n.d.).

<https://www.lucidchart.com/pages/es/que-es-un-modelo-de-base-de-datos#:~:text=El%20modelo%20jer%C3%A1rquico%20organiza%20los,almacenar%20la%20base%20de%20datos.>