Actividad Integral: BST

22 OCTUBRE

Programación de Estructuras de Datos
Adrian Becerra Meza
Profesor: Luis Ricardo Peña Llamas
Grupo: 13

Tecnológico de Monterrey

Binary Search Tree

Un arbol es una estructura de datos jerarquica y no lineal la cual consiste en una colleción de nodos, cada no apunta a sus hijos.

Los arboles tienen cierta terminología:

- Nodo Padre: El nodo predecesor.
- Nodo Hijo: El inmediato sucesor de un nodo.
- Nodo Raiz: El nodo más alto de un arbol, no tiene un nodo padre.
- Grado del nodo: La cuenta total de subarboles.
- Nodo Hoja: Nodo que no tiene hijos.
- Nodo Ancestro: Los nodos predecesores.

Una de las razones por las que podemos usar arboles es porque se quiere guardar información que contara con una jerarquía, los arboles proveen moderado acceso y busqueda

Tipos de Arboles Binarios:

- Arbol Binario Lleno: Si cada nodo tiene 0 o 2 hijos
- Arbol Binario Completo: Si todos los niveles estan completamente llenos excepto el ultimo nivel que debe tener todas las llaves en la parte izquierda.
- Arbol Binario Perfecto: Cuando todos los nodos internos tienen dos hijos y todos los nodos hojas estan en el mismo nivel
- Arbol Binario Balanceado: Si la altura del arbol es O(logN) n siendo el numero de nodos.

Complejidad

Arbol Balanceado

Mejor Caso	Peor Caso
O(1)	O(logN)

Arbol no Balanceado

Mejor Caso	Peor Caso
O(1)	O(n)

Operación	Complejidad
Buscar	O(n)
Insertar	O(n)
Eliminar	O(n)

Podriamos determinar si una red esta infectada, por medio de la cantidad de ingresos fallidos por parte del usuario. Esto se debe a que si alguien nomas fallo en entrar una vez, probablemente no este infectada y sea un simple error por parte de la IP.

Sin embargo si tenemos una IP, con una cantidad extremadamente grande de intentos de ingreso, lo más probable es que esta red este plagada de bots o intentos de hackeo, por lo que podremos considerarla como una red infectada.

Para esto usamos las concurrencias obtenidas en la actividad, así como el arbol binario donde guardamos todas las IPS y sus concurrencias como llaves, ya despues sacamos los datos por medio de un inorder, y podemos obtener las IPS con mayor intentos de iniciar sesión.

Bibliografía

- Rahman, M. (s. f.-b). *PHP 7 Data Structures and Algorithms*. O'Reilly Online Learning. Recuperado 22 de octubre de 2021, de https://www.oreilly.com/library/view/php-7-data/9781786463890/c5319c42-c462-43a1-b33d-d683f3ef7e35.xhtml
- GeeksforGeeks. (2018, 22 noviembre). *Complexity of different operations in Binary tree, Binary Search Tree and AVL tree*. Recuperado 22 de octubre de 2021, de https://www.geeksforgeeks.org/complexity-different-operations-binary-tree-binary-search-tree-avl-tree/