Tecnológico de Monterrey

Campus Monterrey

Programación de estructuras de datos y algoritmos fundamentales

Actividad Integral de BST

Profesor: Dr. Eduardo Arturo Rodríguez Tello

A01571226 Mariano Barberi Garza
A00833623 Iván Alberto Romero Wells

26 de Noviembre de 2022

Iván

Introducción

Los ataques cibernéticos se presentan cada vez con más frecuencia en nuestro mundo digital, en donde, la información de todas las personas conectadas por medio del internet, con el uso de múltiples dispositivos, se pone en riesgo. Un ataque cibernético puede simbolizar un gran riesgo para todos sus afectados, ya que, pone en riesgo la información de los usuarios, abriendo múltiples vulnerabilidades de alto nivel, como acceso a información personal, información bancaria, o “backdoors” que afectan de manera directa a personas y compañías.

El rastreo de los ataques cibernéticos nos permiten, no solo darnos cuenta de que estos mismos suceden, sino que también, nos permiten encontrar un origen y comportamiento del ataque, y dar un seguimiento al mismo. De este modo, el determinar acciones por realizar nos permitirán acabar o disminuir los daños por los ataques.

Marco Teórico

Heap Sort

El heapsort es una técnica utilizada para ordenar conjuntos de datos de una manera eficiente, similar a otras formas de ordenamiento, en donde se consiguen tiempos de ejecución de O(nlogn). Así como una alta eficiencia en espacio de memoria, utilizando apenas el espacio necesario para los elementos del conjunto, O(n). El Heapsort es la base para la implementación de una cola de prioridad, en donde un elemento, encuentra su respectivo

lugar, haciendo uso del heap. De esta manera, nos es posible hacer inserciones y modificaciones en un tiempo logarítmico.

Binary Tree

El Árbol binario es una estructura de datos que nos permite representar un conjunto de datos de manera ordenada y jerarquizada por niveles. En donde cada uno de nuestros elementos, están contenidos en nodos, que dirigen hacia dos otros nodos, uno izquierdo y otro derecho. Un árbol binario completo, llena sus niveles con cada vez más nodos, cada nivel puede contener hasta 2^mnodos, en donde m representa el nivel en cuestión.

Binary Search Tree

Un árbol de búsqueda binaria, es un árbol binario, con la propiedad de que cada nodo, se encuentra ordenado por su valor. Es decir que, a la derecha de un nodo, se encontrarán nodos con un mayor valor, y a la izquierda, nodos con un menor valor. Esto nos deja con la ventaja de poder hacer búsquedas e inserciones dentro del árbol de una manera sumamente eficiente. Dado que, para encontrar un nodo, en un árbol de búsqueda binaria balanceado, nos será posible llegar a tiempos de ejecución de O(log n) para estas operaciones.

Desarrollo

Para el desarrollo de esta actividad, se hizo uso de una cola de prioridad, utilizando el Max Heap para ordenar los registros por sus IP’s. Una vez realizado este ordenamiento, aprovechamos que nuestras IP’s se encontraban agrupadas (y ordenadas) para contabilizar sus apariciones. Estas contabilizaciones, fueron almacenadas en un Árbol de Búsqueda Binaria, dado que nos es de suma utilidad, para posteriormente encontrar las IP’s con más apariciones.

Conclusión

En esta actividad, nos fue posible interactuar con estructuras de datos y técnicas que nos permitieron agilizar procesos de ordenamiento, para ambos casos, el ordenamiento de IP's, y el número de apariciones de registros con dichas IP's. Por esto, considero que fue de suma utilidad conocer ambas estructuras para poder abordar dicha problemática de manera eficiente.

¿Cómo podrías determinar si una red está infectada o no?

La facilidad que nos brinda un árbol de búsqueda binaria para manejar datos ordenados, es de suma importancia en una situación de esta naturaleza, ya que nos permite determinar un orden accesible y ágil en el ordenamiento de nuestros datos.

Dado que nuestra situación problema aborda un tema de infección, en donde las relaciones entre dispositivos, y las peticiones que hacen, determina si un dispositivo está infectado o no. Nos es de gran ayuda utilizar esta estructura de datos, para conocer de donde se origina nuestra infección. Ya que, es posible que no toda nuestra red se encuentre infectada, sino que un subárbol de nuestra red lo pueda estar y de esta manera, es posible que conozcamos el origen de la infección. El problema de esto, es que de esta manera, solamente consideramos relaciones unidireccionales, en donde nodos de un nivel no se pueden comunicar con nodos de un nivel superior, como ocurre en la vida real.

Referencias

Heap Sort - GeeksforGeeks. (2013, March 16). GeeksforGeeks. https://www.geeksforgeeks.org/heap-sort/

Binary Heap - GeeksforGeeks. (2014, November). GeeksforGeeks. https://www.geeksforgeeks.org/binary-heap/

What is a Binary Search Tree? (2022). Educative: Interactive Courses for Software Developers. https://www.educative.io/answers/what-is-a-binary-search-tree

Mariano

Importancia y Eficiencia del uso de BST en Situación Problema

Binary Tree

Árbol ordenado, donde cada nodo tiene como máximo dos hijos, pueden tener cero hijos, un hijo o dos, pero no más. Un “rooted tree” naturalmente tiene niveles que se delimitan por la distancia desde la raíz, por lo que para cada nodo se puede definir una noción de hijos como los nodos conectados a él un nivel por debajo.

Binary Search Tree

Es un “binary tree” donde cada nodo en el árbol del lado izquierdo tiene un valor menor que el de la raíz, y cada nodo en el árbol derecho tiene un valor mayor que la raíz. Creando así una situación donde cada nodo puede ser comparado para saber si es mayor o menor, también se podría ordenar de manera alfabética, pero en este caso donde se nos pide ordenar IP’s primero debemos hacer que la ip tenga un valor numérico. O(n) donde n es la altura del árbol.

IP’s

En esta implementación fue importante darle un valor numérico a las IP’s para poder ordenarlo, la forma para hacer esto es multiplicar 256^[posición] * [valor del punto], por ejemplo si tenemos 190.10.20.40 se multiplicaría 256^0

* 40 + 256^1 * 20 + 256^2 * 10 + 256^3 * 190 = y te daría un valor numérico que se puede ordenar más fácilmente.

Heap Sort

Técnica para clasificar que se basa en comparaciones, basada en la estructura de datos Binary Heap. Se parece a la ordenación por selección en la que primero encontramos el elemento mínimo, se coloca el mismo al principio y repetimos este mismo proceso para los elementos restantes.

¿Cómo podrías determinar si una red está infectada o no?

Hablando sobre la situación problema y de su tema de infección, podemos determinar si un dispositivo está infectado o no a través de las conexiones entre nodos . Nos ayuda bastante utilizar esta estructura de datos (BST), para conocer de dónde viene la probable infección buscada. Ya que, es muy probable que no toda la red de datos esté comprometida por el ataque, sino que puede ser solo algún subárbol de la red y podemos conocer el origen de la infección.

Bibliografía:

Binary search trees. CS 225 | Binary Search Trees. (n.d.). Retrieved November 27, 2022, from https://courses.engr.illinois.edu/cs225/fa2022/resources/bst/

Most common binary tree interview questions & answers [for Freshers & Experienced]. upGrad blog. (2022, November 23). Retrieved November 27, 2022, from https://www.upgrad.com/blog/binary-tree-interview-questions-answers/

Heap sort. GeeksforGeeks. (2022, September 22). Retrieved November 27, 2022, from https://www.geeksforgeeks.org/heap-sort/