



Tecnológico de Monterrey

Programación de estructura de datos y algoritmos fundamentales

Actividad 3.3 Actividad Integral

Sebastián Antonio Almanza A01749694

Tecnológico de Monterrey

4 de Febrero del 2024

Grupo 570

Profesor: Eduardo Arturo Rodriguez Tello

En las ciencias computacionales una de las estructuras de datos de mayor utilidad son los grafos. Un grafo es una estructura de datos de tipo no-lineal que se conforma principalmente por vértices(o nodos) y edges (o arcos), normalmente en un grafo se utilizan los nodos para almacenar información y los arcos para generar conexiones entre uno o más nodos. Para esta actividad se realizó la implementación de un grafo ponderado para almacenar la información de una bitácora que contiene las conexiones entre Ips origen y destino así como sus pesos. Se decidió utilizar esta estructura de datos ya que deseamos saber cuales son las Ips con mayor número de ips adyacentes a la ip de origen, y en base a esto descubrir que Ip podría ser el botmaster de un ataque informático, así como la ip que se encuentre a mayor distancia de esa Ip. El usar un grafo nos permite almacenar en cada nodo las ip que se encuentren en la bitácora y al ser un grafo ponderado cada arco posee el peso de conexiones entre cada ip, además de que con el algoritmo de Dijkstra podemos obtener las distancias entre el botmaster y todas las ip y así conocer la que está más lejos.

Para realizar estas consultas se implementaron los siguientes métodos:

- `shortestPath()`: Este método es el algoritmo Dijkstra que nos permite obtener las distancias entre una Ip dada (en este caso la Ip del botmaster) y todas las demás Ip que se encuentren en el grafo de una forma rápida y eficiente. Este método tiene complejidad $O(n^2)$.
- `consttruyeHeap()`: El método de `construyeHeap()` crea un árbol maxheap que nos ayuda a insertar dentro de él objetos de tipo `infoIp` para al utilizar los métodos de `getTop()` y `pop()` poder obtener las 5 IPs con mayor grado así como todas las demás Ip y sus respectivos grados de salida. La complejidad del método es $O(\log n)$.

En conclusión el implementar un grafo ponderado nos es de utilidad ya que de esta forma podemos obtener la información requerida de forma rápida y eficiente gracias al uso del

algoritmo de Dijkstra y el uso de un max Heap podemos obtener tanto las distancias como las Ips con mayores grados de salida de forma eficaz y pudimos ver como en un escenario real podríamos utilizar estas estructuras para localizar el inicio de un ataque informático.

Referencias

GeeksforGeeks. (2024a, enero 17). *Graph Data structure and Algorithms*.

<https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/>