Instituto Tecnológico y de Estudios Superiores de Monterrey

TC1031.850

Programación de estructuras de datos y algoritmos fundamentales

Actividad

Reflexión actividad integradora 4

Profesor

Eduardo Arturo Rodríguez Tello

Xavier Alfonso Barrera Ruiz

Fecha:

30/05/2023

En este planteamiento, hemos abordado una serie de tareas relacionadas con el análisis de un grafo que representa una red de IPs. A través de la implementación de algoritmos y estructuras de datos, hemos obtenido información útil sobre la red y hemos respondido a preguntas específicas.

Utilizamos una clase Heap para encontrar las 5 IPs con el mayor grado de salida. Esto nos ayuda a extraer eficientemente los elementos máximos de una colección.

Determinamos la dirección IP del boot master utilizando algún criterio adicional o información adicional proporcionada. Esto nos permite identificar la IP que se presume como el nodo central o líder de la red.

Encontramos el camino más corto desde el boot máster a cada una de las otras IPs en el grafo y almacenamos los pares (IP, distancia). Esto nos proporciona información sobre las distancias relativas desde el boot máster a los demás nodos de la red.

Identificamos la dirección IP que en teoria requiere menos esfuerzo para que el boot máster la ataque, basándonos en las distancias calculadas en el paso anterior. Esto nos permite identificar la IP más accesible o cercana al boot master.

Todo esto nos ayuda ya que en la aplicación como por decir, la seguridad nos ayuda a ver los nodos mas vulnerables y esto nos permite mejorar la seguridad, y es fundamental ya que muchas cosas implementan esto un ejemplo son las redes sociales.

Los grafos al final representan una solución a un problema complejo de manera sistematizada y con patrones para resolverlo de la manera más clara posible.

En este caso que es c++ la mayoría de los algoritmos de Dijkstra que usa el método es s O((|V|+|E|)*log(|V|))

Load graph y print graph tienen una complejidad de O(n^2) ya que se recorre la cadena una vez.

FindIpIndex: La complejidad de este método es O(1) ya que solo se busca el indice

En conclusión la complejidad de los métodos es de O(n) o O((|V|+|E|)*log(|V|)) algo muy común en Dijkstra.