



Instituto Tecnológico y de Estudios Superiores de Monterrey

Act 5.2 - Actividad Integral sobre el uso de códigos hash

Programación de estructuras de datos y algoritmos fundamentales (Gpo 570)

Integrantes:

Noreth Sofia Villalpando Saldaña A01368579

Héctor Calderón Reyes A01350637

Campus:

Toluca y Guadalajara

Profesores:

Dr. Eduardo Arturo Rodríguez Tello

Fecha de entrega:

29 de Julio de 2023

Reflexión Hector

El uso de estructuras de datos eficientes es fundamental para la manipulación y análisis de grandes cantidades de información, como es el caso de los registros de bitácora de una red. Las tablas hash, también conocidas como tablas de dispersión, son una de esas estructuras que ofrecen ventajas significativas en términos de eficiencia y velocidad de recuperación de datos. En este contexto, la aplicación de una tabla hash para almacenar y acceder a resúmenes de información de direcciones IP resulta una estrategia acertada.

Las tablas hash funcionan asignando a cada dato una "clave" o "hash", la cual es generada a través de una función de hash. Esta clave se utiliza para determinar el índice en el que se almacenará el dato en la tabla. Cuando la tabla está correctamente dimensionada y la función de hash se ha diseñado de manera efectiva, el tiempo de búsqueda, inserción y eliminación de datos en una tabla hash es constante, es decir, $O(1)$ (Cormen, Leiserson, Rivest, & Stein, 2009). Esta complejidad computacional es ideal para manejar grandes volúmenes de datos, como los generados por las bitácoras de una red.

En el caso de la evidencia, la tabla hash se utiliza para resumir información de cada dirección IP, incluyendo el total de direcciones accesadas desde esa IP y el total de direcciones que intentaron acceder a ella. Este resumen permite una rápida recuperación de la información pertinente para cada IP, facilitando análisis en tiempo real o posteriores.

Es importante notar que el rendimiento de las tablas hash depende en gran medida del manejo de colisiones. Una colisión ocurre cuando dos o más datos resultan en la misma clave hash y, por lo tanto, intentan ocupar el mismo índice en la tabla. El manejo de estas colisiones puede afectar considerablemente la eficiencia de la tabla hash. En el caso, se usa el método de dirección abierta con prueba cuadrática, el cual, aunque efectivo, puede incrementar el tiempo de búsqueda si el número de colisiones es alto (Cormen et al., 2009).

A medida que el número de colisiones aumenta, la eficiencia de las operaciones de la tabla hash se ve comprometida. Idealmente, una tabla hash operaría en tiempo constante ($O(1)$). Sin embargo, si las colisiones son frecuentes, la complejidad computacional puede acercarse a $O(n)$, donde n es el número de elementos almacenados en la tabla (Cormen et al., 2009). Este comportamiento se puede mitigar asegurándose de que la tabla hash esté suficientemente dimensionada para la cantidad de datos que se espera manejar y que la función de hash sea lo suficientemente robusta para minimizar las colisiones.

Por último, es importante destacar que a pesar de los posibles inconvenientes con las colisiones, las tablas hash siguen siendo una de las estructuras de datos más eficientes disponibles. La capacidad de realizar búsquedas, inserciones y eliminaciones en tiempo constante las convierte en una excelente opción para manejar grandes cantidades de datos. Esta actividad es un excelente ejemplo de cómo se pueden aplicar para resolver problemas complejos en el ámbito de las redes y la seguridad informática.

Reflexión Noreth

El análisis de registros de accesos de IPs en el contexto de ciberseguridad es una tarea crítica para identificar posibles amenazas y proteger los sistemas. En esta situación problema, se llevó a cabo la lectura de una bitácora de registros de IPs y su representación en una lista de adyacencias, seguida de la implementación de una tabla hash para almacenar un resumen de la información asociada a cada IP. A través de esta implementación, se logró una organización eficiente y una rápida recuperación de datos, elementos fundamentales en la detección temprana de actividades maliciosas.

En primer lugar, se utilizó una lista de adyacencias para representar el grafo dirigido de las IPs. La complejidad computacional de esta implementación para leer y almacenar los datos es de $O(n + m)$. La representación del grafo de esta manera permite obtener de manera rápida las direcciones accesadas desde una IP específica, lo que es útil en el análisis de patrones de acceso y posibles anomalías.

Luego, se implementó una tabla hash utilizando el método de dirección abierta con prueba cuadrática para almacenar un resumen de la información relevante de cada IP. La complejidad computacional de las operaciones de inserción, búsqueda y eliminación en la tabla hash es $O(1)$ en el mejor caso y $O(n)$ en el peor caso. La eficiencia del método de dirección abierta se basa en la resolución efectiva de colisiones mediante el desplazamiento a posiciones próximas (DSTool).

Las colisiones son inevitables en una tabla hash, especialmente cuando el tamaño de la tabla es pequeño en comparación con la cantidad de elementos a almacenar. El aumento del número de colisiones puede tener un impacto negativo en el rendimiento, ya que se requiere una búsqueda adicional para encontrar una posición libre para insertar la información (EcuRed). Para mitigar este problema, es importante elegir una función hash adecuada y un tamaño de tabla suficientemente grande para reducir la probabilidad de colisiones y mantener un rendimiento óptimo.

La importancia de las tablas hash en esta situación problema radica en su capacidad para organizar y acceder eficientemente a la información relacionada con las IPs. Proporcionan un acceso directo y rápido a los datos, lo que es crucial en el ámbito de la ciberseguridad, donde la detección temprana de actividades maliciosas y la toma de decisiones rápidas son fundamentales para proteger los datos.

Referencias

- *DSTool: Herramienta para la programación con estructuras de datos.* (s.f.).
<http://www.hci.uniovi.es/Products/DSTool/hash/hash-queSon.html>
- EcuRed. (s.f.). *Tabla hash - EcuRed.* https://www.ecured.cu/Tabla_hash
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). The MIT Press. URL: https://pd.daffodilvarsity.edu.bd/course/material/book-430/pdf_content
-
- Zobel, J., & Moffat, A. (1998). Exploring the effectiveness of data compression on the World Wide Web. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (Vol. 5, pp. 4214-4219). IEEE.