

Reflexión actividad integradora #4

La eficiencia en algoritmos de ordenamiento y de búsqueda es una de las cosas más importantes a tener en cuenta al momento de desarrollar un programa que use estos. Teniendo en cuenta esto, se tienen que escoger los algoritmos lo más eficientes posibles para disminuir el tiempo de ejecución y aumentar la eficiencia

Para esta práctica en particular, se utilizaron los grafos como estructura de datos para así llegar a la realización y correcta ejecución de nuestras funciones previstas. Siendo esta una situación que requiere de lectura de IPs y conexiones entre 2 o más direcciones, los Grafos nos permiten una visualización adecuada de la información y sus interconexiones.

Además de esto, los algoritmos de búsqueda que usan los grafos, como el Dijkstra resultan muy útiles para la búsqueda de información eficiente

Es decir que los grafos proporcionan una herramienta poderosa y flexible para analizar las conexiones y relaciones en una red de IPs, permitiendo identificar nodos críticos, analizar patrones de tráfico y optimizar la respuesta a amenazas de seguridad.

Al igual que en la anterior práctica, usar un binary heap sería mucho mejor para el correcto y eficiente funcionamiento del algoritmo para buscar las 5 ips con más accesos, esto debido a su eficiencia de $O(\log n)$, que es útil cuando se están procesando y actualizando dinámicamente las conexiones de las IPs en la red..

Un Binary Heap es preferible para esta tarea porque ofrece una combinación óptima de eficiencia en tiempo de ejecución y simplicidad, especialmente cuando se trata de mantener y acceder repetidamente a las IPs con los mayores grados de salida en una red grande.

Bibliografía

Anzola, G. (2024, 10 enero). *Algoritmos de Ordenamiento: organización de datos, mejor experiencia de búsqueda.* Listopro Community.

<https://community.listopro.com/algoritmos-de-ordenamiento-organizacion-de-datos-mejor-experiencia-de-busqueda/>

Quicksort – numerentur.org. (s. f.).

<https://numerentur.org/quicksort/#:~:text=Quicksort%20es%20un%20algoritmo%20de.no%20es%20el%20m%C3%A1s%20r%C3%A1pido.>

Khandelwal, V. (2023, 25 octubre). *What is Merge Sort Algorithm: How does it work, and More.* Simplilearn.com.

<https://www.simplilearn.com/tutorials/data-structure-tutorial/merge-sort-algorithm#:~:text=with%20practical%20examples.-,What%20is%20a%20Merge%20Sort%20Algorithm%3F,sublists%20into%20a%20sorted%20list.>

GfG. (2024, 18 marzo). *Introduction to Doubly Linked List Data Structure and Algorithm Tutorials.* GeeksforGeeks.

<https://www.geeksforgeeks.org/data-structures/linked-list/doubly-linked-list/>

GeeksforGeeks. (2024, 11 marzo). *Introduction to Max-Heap Data Structure and Algorithm Tutorials.* GeeksforGeeks.

<https://www.geeksforgeeks.org/introduction-to-max-heap-data-structure/>