Valeria Salazar Pinto
A01736652

Reflexión sobre el uso de algoritmos de ordenamiento y búsqueda

Hoy en día prácticamente todo depende de sistemas computacionales. Desde cosas simples como redes sociales hasta temas más delicados como la seguridad digital. Con esto también han aumentado los ataques cibernéticos, muchos de ellos hechos por bots que generan miles de accesos en poco tiempo. Por eso es tan importante poder analizar grandes cantidades de información de manera rápida y eficiente.

En esta actividad se nos presenta una situación relacionada con ataques cibernéticos y el análisis de una bitácora con muchos registros. El reto no es solo guardar los datos, sino poder ordenarlos y buscar información específica dentro de ellos. Para lograrlo, fue necesario usar algoritmos de ordenamiento y búsqueda, y entender por qué algunos funcionan mejor que otros dependiendo del tamaño de los datos.

En esta reflexión voy a hablar sobre la importancia de estos algoritmos, por qué se eligieron, cómo se comportaron en la práctica y cómo se relaciona eso con su complejidad computacional.

Importancia del ordenamiento en esta situación problema

Cuando se trabaja con una bitácora grande, como en el caso de los registros de accesos o posibles ataques, los datos sin ordenar prácticamente no sirven para un análisis serio. Tener la información desordenada hace que cualquier búsqueda sea lenta y poco eficiente.

El ordenamiento permite organizar los registros siguiendo un criterio claro, como la dirección IP o la fecha y hora. Una vez ordenados, es mucho más fácil detectar patrones, por ejemplo si una misma IP aparece muchas veces en un periodo corto de tiempo o si hay rangos de direcciones que se repiten constantemente.

En un contexto real de ciberseguridad, el tiempo es clave. No sirve de mucho detectar un ataque si el análisis tarda demasiado. Por eso el algoritmo de ordenamiento que se elija tiene un impacto directo en la utilidad del sistema.

Algoritmos de ordenamiento utilizados y su justificación

En esta actividad se trabajó con dos algoritmos de ordenamiento diferentes para poder compararlos:

Se trabajó con **Bubble Sort** el cual es un algoritmo más sencillo, de los que suelen enseñarse primero porque son fáciles de entender e implementar. Este tipo de algoritmos tiene una complejidad de $O(n^2)$, lo que significa que conforme crece el número de datos, el tiempo que tarda en ordenar aumenta muy rápido.Este algoritmo funciona bien cuando el número de registros es pequeño, pero al aplicarlo a una bitácora grande se vuelve lento. En la práctica se nota que el programa tarda más en terminar y eso lo hace poco viable para un problema real como el análisis de ataques cibernéticos.

Valeria Salazar Pinto
A01736652

Por otro lado, se utilizó **Quick Sort** como el segundo algoritmo,el cual tiene una complejidad promedio de O(n log n). Aunque su implementación es un poco más compleja, su rendimiento es mucho mejor cuando se trabaja con muchos datos. En las pruebas realizadas, este algoritmo fue claramente más rápido y estable, incluso cuando el tamaño de la bitácora aumentaba.

La comparación entre ambos deja claro que no siempre el algoritmo más simple es la mejor opción. En problemas donde se manejan grandes volúmenes de información, es necesario priorizar la eficiencia.

```
--- Comparaciones y Swaps ---
Bubble Sort:
Comparaciones: 564933691
Swaps: 281563037

Quick Sort:
Comparaciones: 649493
Swaps: 369465
~/workspace$
```

Comparación práctica y relación con la complejidad computacional

Algo importante de esta actividad es que no solo se quedó en la teoría. Al ejecutar ambos algoritmos con los mismos datos, se pudo ver claramente la diferencia en tiempos de ejecución. El algoritmo de complejidad $O(n^2)$ se vuelve cada vez más lento conforme crecen los datos, lo cual coincide con lo que se aprende en teoría.

Por otro lado, el algoritmo de O(n log n) mostró un comportamiento mucho más eficiente y consistente. Esto confirma que la complejidad computacional no es solo un concepto abstracto, sino una herramienta que ayuda a predecir cómo se va a comportar un programa en la vida real.En un sistema de análisis de ataques, elegir un algoritmo con mala complejidad puede hacer que el sistema sea inútil, incluso si funciona correctamente en términos lógicos.

Una vez que los datos están ordenados, el siguiente paso importante es la búsqueda. Buscar información específica en una bitácora desordenada implica revisar registro por registro, lo cual consume mucho tiempo. Sin embargo, cuando los datos están ordenados, se pueden usar algoritmos de búsqueda mucho más eficientes.

Por ejemplo, la búsqueda binaria permite encontrar un registro específico en mucho menos tiempo que una búsqueda lineal. Esto es especialmente útil cuando se necesita localizar rápidamente una IP sospechosa o revisar si cierto patrón aparece en la bitácora.

En conjunto, un buen algoritmo de ordenamiento y uno de búsqueda eficiente hacen que el análisis de datos sea rápido y efectivo, lo cual es fundamental en temas de ciberseguridad.

Reflexión final

Valeria Salazar Pinto
A01736652

Esta actividad me ayudó a entender que los algoritmos no solo sirven para resolver un problema, sino que la forma en la que se resuelve ese problema importa mucho. En situaciones como el análisis de ataques cibernéticos, donde se manejan grandes cantidades de datos, elegir mal un algoritmo puede hacer que todo el sistema pierda sentido.

Comparar diferentes algoritmos de ordenamiento me permitió ver claramente cómo la complejidad computacional afecta el desempeño real de un programa. Aunque algunos algoritmos son más fáciles de implementar, no siempre son la mejor opción cuando se busca eficiencia.

En conclusión, los algoritmos de ordenamiento y búsqueda son fundamentales para poder analizar información de manera rápida y confiable. Entender su comportamiento y complejidad ayuda a tomar mejores decisiones al momento de diseñar soluciones para problemas de la vida real.

Referencias

Berrios, E. (2025, October 4). *Cómo entender la complejidad algorítmica: O(1), O(n), O(n2) y más, explicada con cosas cotidianas*. Medium. https://medium.com/@ermarly/c%C3%B3mo-entender-la-complejidad-algor%C3%ADtmica-o-1-o-n-o-n%C2%B2-y-m%C3%A1s-explicada-con-cosas-cotidianas-478a97757044

Notación Big O y Guía de Complejidad Temporal: Intuición y matemáticas. (2024, July 29). Datacamp.com; DataCamp. https://www.datacamp.com/es/tutorial/big-o-notation-time-complexity

Búsqueda binaria (artículo) | Algoritmos. (n.d.). Khan Academy. https://es.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search

admin. (2022, October 10). *Hackers y ciberdelitos, todos somos vulnerables*. Gaceta UNAM. https://www.gaceta.unam.mx/hackers-y-ciberdelitos-todos-somos-vulnerables/