



Act-Integradora-1

Conceptos básicos y algoritmos fundamentales

Programación de estructuras de datos y algoritmos fundamentales
(Gpo 850)

Profesor

Eduardo Arturo Rodríguez Tello

Nombres:

Federico Melo B. A00833536

24/03/2025

En una situación como la que plantea este curso, donde se habla de ciberataques y redes de bots, los algoritmos de ordenamiento y búsqueda se vuelven herramientas clave. Cuando hay tanta información circulando, como registros de accesos, IPs o patrones de comportamiento, es importante poder analizarla de forma rápida y ordenada. Ahí es donde entra el uso de algoritmos.

Por ejemplo, ordenar los datos (ya sea por fecha, por dirección IP o por tipo de acceso) ayuda a visualizar mejor lo que está pasando. Si un mismo usuario aparece muchas veces en poco tiempo, o si hay un patrón repetitivo, eso ya es una señal. También es más fácil agrupar la información y detectar irregularidades cuando los datos están bien estructurados. Usar algoritmos como Quick Sort o Merge Sort hace este proceso mucho más eficiente, sobre todo cuando se manejan grandes cantidades de datos. (GeeksforGeeks, 2023)

Así mismo, la búsqueda también es súper importante. Si ya tienes una lista ordenada, puedes encontrar lo que buscas más rápido con una búsqueda binaria. Y si usas tablas hash, puedes identificar accesos sospechosos casi al instante, lo cual es muy útil cuando el tiempo es crítico (GeeksforGeeks, 2023). En temas de ciberseguridad, eso puede hacer toda la diferencia.

Este tipo de procesamiento de datos se usa mucho para detectar bots, como los que se mencionan en el artículo de MIT Technology Review, donde explican que aunque se detuvo una botnet muy grande, todavía hay muchísimas activas (Condliffe, 2017). También hay estudios que explican cómo se usan estas estructuras y algoritmos para clasificar accesos y detectar comportamientos anormales en redes (Zhou et al., 2020).

En general, usar bien estos algoritmos no solo te permite detectar accesos maliciosos de forma más eficiente, también te da una ventaja en la forma en que manejas la información. Tener todo organizado y herramientas que te ayuden a analizarlo rápido puede marcar la diferencia.

```
Ejecutando algoritmos de Ordenamiento
QuickSort
Comparaciones : 286667
Swaps : 156930
Tiempo de ejecucion en ms : 31
-----
InsertionSort
Comparaciones : 70200493
Swaps : 70183700
Tiempo de ejecucion en ms : 3729
-----
BubbleSort
Comparaciones : 282508864
Swaps : 70183700
Tiempo de ejecucion en ms : 7729
-----
SelectionSort
Comparaciones : 141262836
Swaps : 16808
Tiempo de ejecucion en ms : 685
-----
SwapSort
Comparaciones : 141262836
Swaps : 70167661
Tiempo de ejecucion en ms : 6837
-----
MergeSort
Comparaciones : 214758
Swaps : 0
Tiempo de ejecucion en ms : 57
-----
MergeSort
Comparaciones: 214758
Swaps: 0
```

Todos los algoritmos de ordenamiento que se utilizan, tienen sus propias cosas buenas y malas. Pero, a pesar de tener más swaps, el QuickSort fue el más rápido. El MergeSort tomó más tiempo que el QuickSort a pesar de no tener swaps. Algoritmos como Insertion, Bubble y Selection Sort fueron muy ineficientes en sus tiempos de ejecución.

Se eligió usar búsqueda binaria porque la bitácora ya está ordenada. Este algoritmo tiene una complejidad de $O(\log n)$ por consulta, lo que lo hace muy eficiente para hacer muchas búsquedas seguidas. Gracias a esto, el programa puede encontrar registros por fecha de forma rápida y sin gastar muchos recursos.

Referencias:

Condliffe, J. (2017). The FBI shut down a huge botnet, but there are plenty more left. MIT Technology Review.
<https://www.technologyreview.com/s/604138/the-fbi-shut-down-a-huge-botnet-but-there-are-plenty-more-left>

GeeksforGeeks. (2023). Sorting Algorithms. <https://www.geeksforgeeks.org/sorting-algorithms/>

GeeksforGeeks. (2023). Data Structures and Algorithms Tutorials. Recuperado de:
<https://www.geeksforgeeks.org/data-structures-and-algorithms-tutorial/>

Zhou, Y., Qian, H., Yan, Z., & Zhang, Y. (2020). A survey on botnet detection techniques. Procedia Computer Science, 167, 465–472. <https://doi.org/10.1016/j.procs.2020.03.273>