

Julián Andrés González Arguello

A00831514

Programación de Estructura de Datos y Algoritmos Fundamentales

Dr. Eduardo Arturo Rodriguez Tello

Enero 18, 2022

Reflexión Individual Act 1.3

Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales

La situación problema de la clase Programación de Estructura de Datos y Algoritmos Fundamentales está relacionada con la guerra de los bots en el área de los ataques cibernéticos, y como estos pueden causar daño ya que son frecuentes y comunes gracias a la tecnología disponible el día de hoy. Al navegar el internet, si no se toman medidas de precaución adecuadas, las computadoras pueden ser infectadas con software maliciosos que infectan las máquinas para hacerlas formar parte de un botnet. Un botnet, proveniente de los términos robot y red en inglés, es una red de computadoras infectadas controladas frecuentemente por ciberdelincuentes para cometer crímenes como ataques de denegación de servicio distribuido, campañas de spam, y fraude. Los botnets son un problema moderno que puede llegar a afectar a cualquier usuario al cometer errores simples, y es importante siempre tenerlos en consideración y buscar como vencerlos.

La Actividad 1.3 trata con una bitácora de texto que contiene los registros de distintos intentos para hackear e infiltrar una base de datos. Para poder leer esta bitácora y manejarla correctamente, se pueden utilizar los algoritmos de ordenamiento y de búsqueda. Los algoritmos de ordenamiento son algoritmos desarrollados para ordenar conjuntos de datos en un orden deseado, mientras que los algoritmos de búsqueda son un conjunto de técnicas usadas para buscar un valor específico dentro del conjunto de datos. Estos algoritmos son cruciales en el análisis de la bitácora ya que sería sumamente difícil ordenar manualmente los datos y después buscar entre todas las fechas específicas para así combatir los botnets. Existen diferentes tipos de algoritmos que se pueden utilizar para la tarea deseada, y cada uno tiene una complejidad temporal que afecta su eficiencia. En nuestro caso, como algoritmos de ordenamiento usamos el

método Burbuja y el método Merge Sort. El método Burbuja es un algoritmo de ordenamiento que es utilizado para ordenar un vector de forma ascendente comparando elementos de la lista seleccionada e intercambiándolos de lugar, y van van subiendo en el orden como burbujas. Este tiene una complejidad temporal de $O(n^2)$ en su caso promedio al igual que en su peor caso, significando que no es muy eficiente para trabajar con cantidades grandes de información. El método Merge Sort es también un algoritmo de ordenamiento pero este funciona al recursivamente ordenar el arreglo de forma ascendente dividiendo los datos en sublistas de menor tamaño, fusionándose en una sola lista.. Este método tiene una complejidad promedio de $O(n \log n)$ en todos sus casos. Al correr el código se puede ver que el método es Merge Sort es más eficiente porque ordena el vector a una velocidad más rápida. Al dividir el vector en sublistas de un tamaño menor y usar estas para comparar los valores, el algoritmo reduce el tiempo necesario para correrlo. Como algoritmo de búsqueda usamos el método de la Búsqueda Binaria, que funciona al tener una llave y compararla con las dos mitades del vector sucesivamente. Seleccionamos este algoritmo ya que es el más eficiente, teniendo una complejidad temporal de $O(\log_2 n)$ en su caso promedio y $O(\log n)$ en su peor caso.

Referencias

Hanna, K. (s.f). Botnet. TechTarget. Recuperado el 18 de enero del 2023 de:

https://www.techtarget.com/searchsecurity/definition/botnet?track=NL-1823&ad=931942&src=931942&asrc=EM_NLN_122529935&utm_medium=EM&utm_source=NLN&utm_campaign=20200121_Word+of+the+Day%253A+botnet

Rodriguez, E. A. (s.f.). Algoritmos de Ordenamiento, part 3. Recuperado el 18 de enero del 2023 de: <https://experiencia21.tec.mx/courses/328618/files/123360238?wrap=1>

Rodriguez, E. A. (s.f.). Algoritmos de Búsqueda. Recuperado el 18 de enero del 2023 de: <https://experiencia21.tec.mx/courses/328618/files/123360238?wrap=1>