Paulina Lizet Gutiérrez Amezcua
A01639948
26 de noviembre del 2021
Programación de estructuras de datos y algoritmos fundamentales

Act 6.2
Reflexión Final de Actividades Integradoras de la Unidad de Formación TC1031

A lo largo del curso desarrolle diferentes actividades que me ayudaron a la solución de las actividades integradoras, al realizarlas llegue a la conclusión que la más eficientes son las que usan apuntadores ya que contienen la dirección de memoria de otra variable lo que hace que sea más eficiente. Las actividades que considero más eficientes fueron:

El árbol de búsqueda binaria, el cuál es un método simple y dinámico, donde cada nodo tiene una clave comparable y satisface la restricción de que la clave en cualquier nodo es más grande que las claves en todos los nodos en el subárbol izquierdo de ese nodo y más pequeña que las claves en todos los nodos del subárbol derecho de ese nodo. Aumenta la eficiencia de la operación de búsqueda en un árbol (logarítmico) al igual que en la búsqueda binaria, admiten todo lo que puede obtener de una matriz ordenada: búsqueda eficiente, recorrido hacia adelante / hacia atrás en orden desde cualquier elemento dado, búsqueda de elemento predecesor / sucesor y consultas de máximo / mínimo, con el beneficio adicional de inserciones y eliminaciones eficientes. Su complejidad de tiempo es O (h) donde h es la altura del árbol.

Las listas doblemente ligadas, las cuales son una variación de la lista enlazada en la que la navegación es posible en ambos sentidos, ya sea hacia adelante y hacia atrás fácilmente en comparación con la lista enlazada única. Son muy eficientes por su facilidad de recorrido, permite todo tipo de codificación que se le agregue como métodos de inserción o eliminación y facilita la búsqueda de datos, permite almacenar datos de manera más organizada, hace más rápida la búsqueda porque ya no necesita referenciar el elemento

anterior. Su complejidad computacional para eliminar o insertar un elemento, ya sea al inicio o al final, es O(1) y para buscar o acceder es O(n).

Aunque las dos soluciones son eficientes, considero que la mejor es la del árbol binario debido a que cada actividad tenía una gran cantidad de datos y en este caso su complejidad es O (h) donde h es la altura del árbol y para las listas doblemente ligadas es O(n), por lo tanto h es menor que n en el caso de la gran cantidad de datos. De igual manera es la manera más eficiente para resolver la situación problema ya que permite búsquedas optimizadas para la detección oportuna de accesos maliciosos a través de redes de robots, esto tomando como ejemplo las ip´s de la actividad, podríamos obtener las ip´s que más acceso tuvieron en menos tiempo lo que significaría que es un bot y por lo tanto está infectada.

Para la primera entrega de la situación problema utilizamos algoritmos de ordenamiento y búsqueda, con mi equipo aplicamos el método de burbuja y búsqueda secuencial pero al realizar las búsquedas el programa tardaba mucho tiempo en compilar por lo que llegamos a la conclusión de que podríamos mejorar los métodos para hacerlo más efectivo al aplicar el método de Merge Sort ya que parte el vector en mitades y las ordena lo que hace que sea un proceso más rápido y para la búsqueda podríamos mejorar a utilizar una búsqueda binaria que compara el elemento que sé está buscando con el central y si coinciden finaliza, si no busca si el elemento es mayor o menor y se hace la búsqueda en le arreglo correspondiente, lo que hace mucho más rápido el proceso y para esta situación problema en la que se esta trabajando con muchos datos es necesario tener algoritmos que eficienten la búsqueda.

Para la segunda evidencia se utilizo el mismo código que en la primera pero ahora el programa buscaba las ips que más se repetían, este proceso seguía siendo lento pero decidimos no usar las listas doblemente ligadas para no cambiar la estructura del código que ya teníamos y no complicarnos tanto, pero al hacer la reflexión nos dimos cuenta que hubiera sido de gran utilidad utilizar las listas doblemente ligadas ya que facilitaría el recorrido, la búsqueda de datos y nos permitiría agregar el método que deseemos.

Link a la carpeta de drive con las actividades integradoras: https://drive.google.com/drive/folders/1yNuYaXIARpasBTOlO6PZLHRfUumFIiYy?usp=sharing

Link del repositorio: https://github.com/PaulinaGtz/TC1031.13