Luis Enrique Gutierrez

A00837481

Programación de Estructura de Datos y algoritmos Fundamentales (Gpo 850)

Profesor:

Dr. Eduardo Rodriguez Tello

10 de Junio del 2024

Reflexión Actividad Integradoras

Resumen y Reflexión

Experiencia y Aprendizaje

Durante el transcurso de este curso, se introdujo a todo tipo de estructuras de datos, desde las más básicas hasta las que se pueden considerar como las más complejas. En cada actividad realizada, se implementó una nueva estructura que demostraba ser más eficiente que la anterior. Lo anterior lo sabíamos en base a la complejidad computacional de cada algoritmo al igual al realizar una comparación entre pros y contras de cada uno. En este ensayo se realizará un resumen breve de cada una de las actividades al igual de lo aprendido y conclusiones.

Algoritmos de Ordenamiento (1):

La primera actividad del curso se relaciona a algoritmos de ordenamiento. En esta se nos pidió hacer un algoritmo que ordenara una serie de datos. Los algoritmos para poder llevar a cabo la actividad fueron BubbleSort y QuickSort. Al realizar una serie de pruebas de ambos algoritmos se llegó a la conclusión que QuickSort es el algoritmo más efectivo al manejar grandes listas de datos, confirmándose con la complejidad computacional O(nlogn). BubbleSort, a diferencia de QuickSort tiene una complejidad de O(n^2) por lo que no es eficiente al manejar grandes cantidades de información.

Estructuras Lineales (2):

En cuanto a la segunda actividad integradora, se introdujeron las estructuras de datos lineales, en este caso las listas doblemente ligadas. Demostraron ser una estructura muy volátil por su característica de acceso bidireccional de los nodos. Esto facilita el poder insertar y eliminar datos, teniendo una complejidad O(1). Se vio que fue efectivo para poder manipular grandes cantidades de información a diferencia de utilizar otros tipos (como vectores o arreglos).

Arboles Binarios (3):

Seguidamente, se estuvo trabajando con árboles binarios(BST), primera estructura no lineal vista. Fue una estructura efectiva para realizar operaciones de búsqueda, más aún cuando hay de por medio una cantidad considerable de datos e información. Tiene una complejidad computacional en promedio de O(log n) pero puede verse afectada y pasar a ser una complejidad de O(n).

Grafos (4):

Los grafos, otra estructura no lineal, se pueden describir como si fuera una telaraña en la cual hay muchos puntos unidos. Cada punto contiene información y se puede relacionar con uno o más puntos (nodos). Se empleó el método de Dijkstra, que nos permite encontrar

rutas más cortas para llegar a cierto nodo. Con una complejidad de O((n+m)logn), y al combinarlo con binary heap fue un método efectivo para y perfectamente moldeable para realizar cualquier tipo de tarea.

Tablas Hash (5):

Esta última estructura, se caracterizó por ser muy eficiente para realizar operaciones de inserción y eliminación con una complejidad promedio de O(1). Al estar agregando información a la tabla se le asignaba una clave a cada dato. En este contexto se agregó a dicha tabla mucha información por lo que podrían surgir colisiones entre elementos. Si se realiza un buen manejo de métodos eficientes que reduzcan el valor de carga y así evitar inconvenientes de este tipo.

Para saber cual es el algoritmo empleado más efectivo, una herramienta que nos ha servido es realizar el análisis de complejidad computacional. Se sabe que los algoritmos más efectivos son aquellos que tienen complejidad de O(logn) o O(1). En base a eso los algoritmos que cumplen esa condición son las tablas hash (O(1)) o bien los árboles binarios (O(logn)). Así se demostró en los experimentos realizados durante las entregas de las actividades integradoras.

Siempre se puede hacer algún cambio que mejora la efectividad de algún algoritmo, es por eso que tenemos que tomar en cuenta los pros y contras del cambio. Por ejemplo, un cambio que haría sería cambiar el BubbleSort de la primera actividad por un MergeSort. Tomando en cuenta las complejidades computacionales de ambos algoritmos, la complejidad de MergeSort, O(nlogn), es mucho más eficiente que la de BubbleSort por lo que podríamos tener un mejor análisis y comparación con el QuickSort. Otro cambio posible sería reemplazar las estructuras lineales por otras no lineales como lo puede ser grafos. Ya que se ha visto que los grafos son muy poderosos y más eficientes que estructuras lineales para el manejo de grandes cantidades de información (Ips) o incluso para detectar un ataque cibernético.

Al realizar todas las actividades anteriormente mencionadas pude desarrollar las diferentes competencias del curso. Evaluación del problema, Toma de decisiones, Implementación de acciones fueron las competencias que trabajé al máximo. ¿Cómo lo hice? Al hacer un análisis de lo que se estaba pidiendo, basándose en conocimiento e investigación logre tomar la ruta correcta para actuar. Así como fue el saber cual complejidad era la más efectiva para poder llegar a implementar el algoritmo correspondiente.

En conclusión fue una unidad de formación en la cual aprendí mucho, sobre todo para relacionarlo con problemas de la vida cotidiana y cómo funcionan ciertas cosas tecnológicas.

La única desventaja que puedo ver es la modalidad en línea, porque en lo personal me cuesta trabajo concentrarme y ahora fue cuando más me afectó. Pero dejando eso a un lado pude sacarle mucho provecho. Ahora tengo más conocimientos sobre algoritmos que me llegaran a servir en el futuro.