

Reflexiones Estructuras de Datos

ACT 1.3; fue difícil implementar tecnologías en un problema que te da cualquier solución posible. Lo más difícil de esta actividad fue encontrar la solución más eficiente y más fácil de implementar. Como no se ha utilizado ninguna estructura de datos o algoritmo específico el código no es muy eficiente, esto en retrospectiva muestra la importancia de aprender de las estructuras y los diferentes algoritmos que hay para manejar y manipular información.

ACT 2.3 La actividad trata sobre datos lineales, en esta actividad podemos ver cómo se pueden utilizar vectores para resolver problemas aunque no sea la manera más eficiente. Una estructura de datos lineales tienen como función crear una secuencia, añadir un elemento, borrar un elemento, consultar un elemento y por último borrar la secuencia de información. Es una estructura de datos muy sencilla lo cual significa que es muy fácil de implementar aunque a escala más grande es más lenta y menos eficiente que muchas otras.

ACT 3.4 trata sobre Árboles binarios, los árboles binarios son listas enlazadas con nodos padres y nodos hijos que se relacionan entre sí para crear un árbol. Los árboles binarios son muy útiles para mantener información en listas vinculadas entre sí. Esto permite hacer búsquedas muy rápidas llamadas BST o Binary Search Tree y encontrar información que se relaciona entre sí.

ACT 4.3 Trata sobre los grafos. Los grafos son una estructura parecida a árboles solo que no son lineales. Los grafos están formados por nodos y vértices. Hay dos tipos de grafos los dirigidos y los no dirigidos y ambos se utilizan para diferentes situaciones. Los grafos son utilizados para crear datos con conexiones entre sí lo cual ocasiona que se puedan encontrar datos de una manera sencilla. Esto nos permite estudiar la relación de datos y manipuladores de la mejor manera. Los grafos se ven muy comúnmente en ver cómo funciona el internet y las relaciones que hay en todo el mundo. Algunos ejemplos son, carreteras, trenes, internet y más.

ACT 5.2 La Tabla Hash es una estructura de datos que almacena datos de forma asociativa. En una tabla hash, los datos se almacenan en un formato de matriz, donde cada valor de datos tiene su propio valor de índice único. El acceso a los datos es muy rápido si conocemos el índice de los datos deseados. Las tablas hash son muy importantes para la organización y manejo de información en algoritmos ya que son capaces de mantener la información guardada en llaves y diferentes valores. Esto permite acceder fácilmente a estos valores ya que las llaves son únicas. El problema más grande de las tablas Hash son las complicaciones al duplicarse y repetirse.

En mi opinión no hay una estructura de datos que sea completamente mejor que otra ya que se requieren diferentes soluciones a los problemas y cada problema puede necesitar una estructura diferente. Lo más importante de esta clase además de entender a utilizar las estructuras de datos y sus algoritmos también es aprender a saber cuando utilizarlos y saber cómo resolver problemas con estas estructuras. Si tuviera que elegir alguna estructura de datos en específico escogería las tablas hash para almacenar valores con llaves y que se pueda encontrar fácilmente. Utiliza los árboles binarios para guardar valores que dependen de algunos valores. Por último utilizará los grafos para almacenar datos que están relacionados entre sí y que parezca una red.

De las actividades anteriores se que podría mejorar las primeras 3 actividades ya que en el momento no entendía muy bien las estructuras lo cual significa que los códigos que utilice no fueron muy eficientes. Los algoritmos de búsqueda y para obtener la información de los archivos no fueron muy buenos entonces mejoraría esos algoritmos y las estructuras. Al acabar las últimas actividades aprendí mucho sobre cómo se implementa el código en c++ utilizando las estructuras y sus algoritmos. De esta manera pudimos aprender a utilizar la información de la mejor manera y llegar a soluciones eficientes y correctas para cada problema.