1-¿Clasificar las estructuras de Datos y determinar cuál es la solución que prevé cada una?

* Arreglos (Arrays):

Solución: Los arreglos ofrecen una manera eficiente de almacenar y acceder a elementos del mismo tipo de datos en una secuencia continua. Son útiles cuando se necesita acceder a elementos por índice o realizar operaciones matemáticas en conjuntos de datos.

* Listas enlazadas (Linked Lists):

Solución: Las listas enlazadas permiten almacenar elementos en nodos que están conectados mediante punteros. Proporcionan flexibilidad en la inserción y eliminación de elementos, pero el acceso aleatorio es menos eficiente. Son útiles cuando se requiere una estructura de datos dinámica que permita agregar o eliminar elementos de manera eficiente.

* Pilas (Stacks):

Solución: Las pilas siguen el principio LIFO (Last In, First Out) y se utilizan para almacenar elementos y acceder a ellos en un orden específico. Proporcionan operaciones como "apilar" (push) para agregar elementos y "desapilar" (pop) para eliminar elementos. Son útiles en situaciones donde se necesita un seguimiento temporal de las operaciones o en la evaluación de expresiones matemáticas.

* Colas (Queues):

Solución: Las colas siguen el principio FIFO (First In, First Out) y se utilizan para almacenar elementos y acceder a ellos en un orden específico. Proporcionan operaciones como "encolar" (enqueue) para agregar elementos y "desencolar" (dequeue) para eliminar elementos. Son útiles en situaciones donde se requiere un procesamiento en orden secuencial, como el procesamiento de tareas en un sistema.

* Árboles (Trees):

Solución: Los árboles son estructuras jerárquicas que consisten en nodos conectados mediante enlaces. Proporcionan una manera eficiente de organizar y buscar datos jerárquicos. Los diferentes tipos de árboles, como árboles binarios, árboles de búsqueda binaria o árboles AVL, brindan soluciones específicas para diferentes requisitos, como búsquedas eficientes, ordenamiento y almacenamiento jerárquico.

* Grafos (Graphs):

Solución: Los grafos son estructuras que representan relaciones entre elementos mediante nodos y aristas. Son útiles para modelar situaciones complejas donde los elementos están interconectados. Los grafos permiten resolver problemas como la búsqueda de caminos, la representación de redes sociales, la planificación de rutas, entre otros.

2-¿Explicar el concepto que está detrás de “BigO”, dar 3 ejemplos?

El concepto "Big O" es una notación utilizada en ciencias de la computación para describir la eficiencia o complejidad de un algoritmo. Indica cómo el tiempo de ejecución o el uso de recursos (como el tiempo o el espacio) crece en relación con el tamaño de entrada del problema.

tres ejemplos comunes de notación Big O:

* O(1) - Constante: Este caso indica que el tiempo o el uso de recursos del algoritmo no depende del tamaño de entrada. El algoritmo tiene una eficiencia constante y ejecuta una cantidad fija de operaciones, sin importar cuánto aumente el tamaño del problema. Por ejemplo:
  + Acceder a un elemento en un arreglo por índice.
* O(n) - Lineal: Este caso indica que el tiempo o el uso de recursos crece de manera proporcional al tamaño de entrada. A medida que el tamaño del problema aumenta, el algoritmo realizará una cantidad de operaciones en proporción directa al tamaño de entrada. Por ejemplo:
  + Recorrer una lista enlazada o un arreglo.
* O(n^2) - Cuadrático: Este caso indica que el tiempo o el uso de recursos crece en función del cuadrado del tamaño de entrada. A medida que el tamaño del problema aumenta, el algoritmo realizará una cantidad de operaciones que es el cuadrado del tamaño de entrada. Por ejemplo:
  + Recorrer una matriz bidimensional.

3-¿Dar ejemplos de uso de Tkinter junto a SQLite en casos de Apps?

-Aplicación de gestión de inventario: Aplicación de gestión de inventario utilizando Tkinter para la interfaz de usuario y SQLite para almacenar la información del inventario. La interfaz de usuario puede incluir campos de entrada para agregar nuevos productos, una tabla que muestra los productos existentes y botones para editar, eliminar o buscar productos.

-Aplicación de diario personal: Una aplicación de diario personal utilizando Tkinter para crear la interfaz de usuario y SQLite para almacenar las entradas del diario.