
TDA y visualización de datos con Graphviz

202044192 – Elian Angel Fernando Reyes Yac

Resumen

Las listas enlazadas son estructuras de datos conformadas por nodos. Los nodos son un conjunto de secuencias de datos que se guardan de manera arbitraria los cuales son referenciados por medio de punteros. Los nodos pueden tener punteros que apunten al nodo anterior y/o posterior.

Los archivos xml se han vuelto cada vez más utilizados debido a su fácil lectura e interpretación. Al igual que su fácil implementación de programación orientada a objetos.

Al trapazar con listas enlazadas se puede volver un poco complejo para el usuario comprender el flujo y la relación de los nodos por lo cual se decidió utilizar una herramienta que ayudara a mostrar gráficamente las matrices, dicha herramienta es Graphviz.

Graphviz permite una mejor comprensión grafica de las listas enlazadas y los nodos que la componen, volviendo el proyecto mucho más amigable para el usuario final.

Palabras clave

- Listas enlazadas
- Graphviz
- Documentos XML
- POO

Abstract

Linked lists are data structures consisting of nodes. Nodes are a set of arbitrarily stored data sequences that are referenced by pointers. Nodes can have pointers pointing to the previous and/or next node.

The xml files have become increasingly used due to their easy readability and interpretability. As well as their easy implementation of object oriented programming.

When plotting with linked lists it can become a bit complex for the user to understand the flow and relationship of the nodes so it was decided to use a tool that helps to graphically display the matrices, this tool is Graphviz.

Graphviz allows a better graphical understanding of the linked lists and the nodes that compose it, making the project much more friendly for the end user.

Keywords

- Linked Lists
- Graphviz
- XML document.
- POO

Introducción

Se nos ha pedido realizar un programa para comprimir señales de audio conformadas por tiempo, amplitud, y frecuencia. Para poder dar solución al problema planteado se utilizó listas enlazadas para formar matrices y así poder analizar los datos de mejor manera. Para comprimir las señales se convierte la matriz original en una matriz de patrones formada por ceros y unos, donde uno es cualquier frecuencia diferente a 0. Con la matriz de patrones se crea una matriz reducida donde las filas que tengan el mismo patrón de ceros y unos se suman, sumando las frecuencias de la matriz original.

El programa se desarrolló en el lenguaje de Python, el programa cuenta con una interfaz simple que se muestra por consola. Dicho programa procesa un archivo xml proporcionado por el usuario, pudiendo realizar graficas de las diferentes matrices antes mencionadas, y un archivo xml con la matriz reducida.

Desarrollo del tema

Se nos ha pedido un programa para poder procesar señales de audio que serán dadas en un archivo xml, para luego comprimir las señales con los mismos patrones.

Para comenzar con la resolución del problema hay que considerar que el archivo de entrada es un archivo de tipo xml, el cual se compone de diferentes etiquetas las cuales crean niveles de jerarquía. Las etiquetas del archivo xml que procesara el programa tienen nombres que hacen alusión al dato que contiene, es por ello que su lectura es relativamente sencilla. La etiqueta principal es “senales” la cual contiene toda la información del archivo. La siguiente etiqueta es “senal” la cual tiene como atributos el

nombre, tiempo, y amplitud que tendrán las sub etiquetas. La sub etiqueta es “dato” la cual tiene el atributo tiempo y amplitud, dicha etiqueta contiene el valor de la frecuencia que se encuentra en las coordenadas dadas por los atributos tiempo y amplitud. Siendo tiempo las filas y amplitud las columnas.

Entendiendo la estructura del xml podemos inferir que se puede crear una matriz utilizando listas enlazadas.

a. Menú:

El menú se muestra por consola, el cual debe de ser simple y de fácil comprensión para el usuario. Conteniendo las diferentes opciones que se nos fueron pedidas, de esta manera el menú principal tiene las siguientes opciones:

Menú principal:

1. Cargar archivo
2. Procesar archivo
3. Escribir archivo salida
4. Mostrar datos del estudiante
5. Generar gráfica
6. Inicializar sistema
7. Salida

Figura 1. Menú principal.

Fuente: Universidad de san Carlos de Guatemala, enunciado Proyecto 01, página 07.

Analizando cada un de las opciones y tomando en cuenta que el usuario puede cometer errores al momento de trabajar con el programa se a implementado las sentencias “try except” para asi en caso de que el usuario no realice algo adecuado en el programa este siga funcionando, imprimiendo por consola el posible error que ocurrió.

A continuación, se explica de manera resumida que hace cada opción del menú.

Cargar archivo: Esta opción despliega una ventana desde la cual el usuario debe de escoger un archivo xml para posteriormente procesar.

Procesar archivo: como su nombre indica esta opción procesa el archivo antes cargado al programa, imprimiendo por consola lo que se esta realizando lo cual incluye las señales identificadas, las matrices de patrones realizadas, los grupos y las matrices reducidas.

Escribir archivo de salida: permite al usuario escoger mediante una ventana donde desea guardar el archivo de salida en formato xml que contiene la matriz de señales reducidas.

Mostrar datos del estudiante: muestra en pantalla los datos del estudiante que realizo el programa.

Generar gráfica: permite al usuario escoger mediante una ventana donde desea guardar las graficas de la matriz original, y la matriz reducida.

Inicializar sistema: esta opción vuelve a iniciar el sistema, borrando toda la información procesada con anterioridad.

Al utilizar el programa siempre primero se debe de cargar el archivo, y luego procesar el archivo, si no se realiza de esta manera el programa mostrara por consola que el archivo no fue procesado o no fue cargado. Esto debido a si no se procesa el archivo no se puede generar las graficas o crear el archivo de salida.

b. Manejo del archivo XML:

Como se mencionó anteriormente la estructura del Archivo xml es simple de entender, para ayudarnos A leer y escribir archivos xml existen diferentes librerías en Python, a continuación, se presentan algunas de esas librerías.

Tabla I.

Librería para la lectura de archivos xml

FROM	IMPORT
Xml.dom	minidom

Fuente: elaboración propia.

Tabla II.

Librería para la lectura de archivos xml

FROM	AS
Xml.etree.ElementTree	ET

Fuente: elaboración propia.

Con la lectura del archivo xml podemos llenar las listas enlazadas y sus respectivos nodos, cada nodo esta formado por un dato y un apuntador al siguiente nodo, por su parte cada dato esta conformado por tiempo, amplitud, y frecuencia.

Con los nodos ya creados y enlazados podremos crear las listas necesarias para guardar las señales que contienen el archivo xml. Al hacer uso del programa es necesario cargar un archivo antes de poder crear el archivo de salida o generar las gráficas, en caso de que no se cargue ningún archivo y se intente generar el archivo de salida o crear las graficas se mostrara en consola un error indicando que no se ha cargado ningún archivo.

c. Procesar archivos y listas enlazadas.

Procesar el archivo xml es la parte más importante, ya que como se menciono antes si no se pasa por este proceso no se podrá generar las graficas ni el archivo de salida con las señales reducidas.

En este proceso se inicia leyendo el archivo xml y creando la lista enlazada simple llamada “matriz de frecuencia”, posteriormente se crea la “matriz de patrones” donde cualquier frecuencia que no sea

cero se vuelve un numero uno, creando así una matriz conformada únicamente por frecuencias de valores ceros y uno. Con esta matriz creada se crean grupos de las filas que tengan el mismo patrón, para luego sumar su frecuencia de la “matriz de frecuencia” creando así la “matriz reducida” conformada por los grupos que tengan el mismo patrón y como valor de frecuencia la suma de las frecuencias de las filas.

d. Graphviz

Al trabajar con listas enlazadas puede resultar difícil comprender dichas listas solo viendo la lista por consola es por ello que resulta necesario implementar una representación grafica de las listas enlazadas que sea más de entender para el usuario, es en este proceso que herramientas como Graphviz.

Con Graphviz podemos generar graficas fácilmente, esto debido a que en su pagina se puede encontrar diferentes diseños de graficas junto a su respectivo código de Python. El uso de Graphviz con las listas enlazadas resulta simple si se utilizan ciclos para recorrer las listas y así generar los archivos dot que almacenan la información de la gráfica.

Teniendo el archivo .dot generado con la ayuda de Graphviz se puede realizar la transformación a un formato de archivo más cómodo para el usuario, en este caso el formato escogido fue png.

e. Ventajas y desventajas de las listas enlazadas

Las listas enlazadas son una estructura de datos que tiene sus propias ventajas y desventajas. Por un lado, ofrecen flexibilidad en la inserción y eliminación de elementos, lo que las hace ideales para situaciones donde se necesitan cambios frecuentes en la estructura de datos. Además, utilizan memoria de manera eficiente, ya que solo requieren espacio adicional para almacenar punteros. Su tamaño puede crecer o disminuir dinámicamente, adaptándose a las

necesidades del programa. De igual manera existen diferentes tipos de listas enlazadas que permiten adaptar la estructura.

Sin embargo, las listas enlazadas tienen algunas desventajas importantes. El acceso a elementos es generalmente más lento que en los arrays, ya que requiere seguir punteros desde el inicio de la lista, lo cual puede afectar el rendimiento en aplicaciones que necesitan acceso rápido y frecuente a datos.

De la misma manera las listas enlazadas consumen más memoria debido a la gran cantidad de punteros, y su implementación es más compleja en comparación con estructuras de datos más simples. Dependiendo del programa no es la elección más optima.

En última instancia, la elección entre listas enlazadas y otras estructuras depende de las necesidades específicas de la aplicación y las operaciones que se deben realizar con los datos.

Conclusiones

- Las listas enlazadas son una estructura de datos altamente flexible que permite la inserción y eliminación eficiente de elementos en tiempo constante.
- Los archivos XML ofrecen una forma versátil de organizar datos, lo que es fundamental en aplicaciones que requieren una estructura de datos jerárquica y legible.
- Por otro lado, los archivos XML son ideales para representar datos que necesitan ser compartidos entre diferentes sistemas o plataformas.

- Por otro lado los archivos XML, aunque versátiles, pueden volverse complejos y pesados en términos de procesamiento y almacenamiento cuando se utilizan para representar grandes conjuntos de datos debido a las etiquetas repetitivas y la anidación profunda

- Aditya, R. (2022, abril 12). Lista enlazada en Python. Delft Stack.
<https://www.delftstack.com/es/howto/python/linked-list-in-python/>

Anexos:

. **Diagrama de clases:** diagrama que muestra el flujo que tiene la aplicación

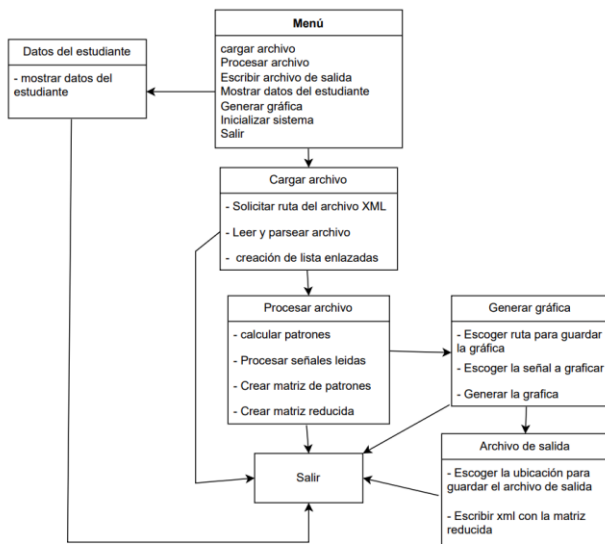


Figura 2. Diagrama de clases.
Fuente: elaboración propia 2023

Referencias bibliográficas

- Amazon. (s/f). *¿Qué es XML?* Amazon.com. Recuperado el 6 de septiembre de 2023, de [https://aws.amazon.com/es/what-is/xml/#:~:text=An%20Extensible%20Markup%20Language%20\(XML,Online%20XML%20editors](https://aws.amazon.com/es/what-is/xml/#:~:text=An%20Extensible%20Markup%20Language%20(XML,Online%20XML%20editors)