Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2021.

PROYECTO 1 – TDA's

201800678 – Wilmer Estuardo Vásquez Raxón

Resumen

Consiste en analizar un archivo XML con una estructura definida, almacenando en memoria el contenido de este para su posterior análisis y procesamiento. Este archivo será el que se procese posteriormente y el que definirá los grafos que se podrán construir. El programa cuenta con diferentes funcionalidades que permitirán reducir los datos para menor uso de memoria.

Es posible imprimir los datos reducidos en un archivo XML en una ruta dada por el usuario. Es posible la creación de grafos generados para una matriz dada la cual se mostrará automáticamente.

Se utilizaron librerías como ElementTree y graphviz las cuales facilitan la lectura y construcción de los xml y los grafos.

El usuario podrá generar el archivo xml cuantas veces quiera siempre que ingrese un archivo de entrada y los grafos se pueden generar cuantas veces se quiera

Palabras clave

XML: archivo con extensión .xml el cual se estructura con etiquetas anidadas las cuales pueden tener atributos

Grafo: figuras referentes a mapas de interconexión de datos.

Librerías: conjunto de archivos que se utilizan para facilitar la programación, a los que llamamos al principio de la página

Abstract

It consists of analyzing an XML file with a defined structure, storing its contents in memory for later analysis and processing. This file will be the one that will be processed later and the one that will define the graphs that can be built. The program has different functionalities that will allow to reduce the data for less memory usage.

It is possible to print the reduced data in an XML file in a path given by the user. It is possible to create graphs generated for a given matrix which will be displayed automatically.

Libraries such as ElementTree and graphviz were used to facilitate the reading and construction of xml and graphs.

The user can generate the xml file as many times as the user wants if the user enters an input file, and the graphs can be generated as many times as desired.

Keywords

XML: file with .xml extension which is structured with nested tags which may have attributes.

Graph: figures referring to data interconnection maps.

Libraries: set of files used to facilitate programming, which we call at the top of the page.

Introducción

Se busca disminuir el espacio en memoria que consumen los datos haciendo que los datos que contienen una estructura similar sus tuplas se sumen para consumir menos memoria.

Se intenta dejar claro y de forma simple el manejo de diferentes datos como son las tuplas para su análisis optimo, y su operación simple para reducción de tuplas.

Se implementan librerías para procesar y escribir archivos XML y PDF para los reportes respectivos los cuales se espera que abran automáticamente.

Desarrollo del tema

Se implementaron TDA's los cuales almacenan todo el contenido obtenido de los archivos, se manejaron ciclos que permiten que el programa siga en ejecución después de realizar cualquier acción.

Se implemento la matriz con TDA's anidados para evitar el uso de vectores, definiendo otro TDA el cual contiene las matrices implementadas.

Para evitar la complejidad de las clases se implementaron varios métodos y funciones en cada TDA para hacerlos mas simples de operar y de acceder a ellos.

Se accedió al contenido del xml con la librería ElementTree y se obtuvo cada matriz y se itero en ella para obtener cada valor y sus atributos.

Se implemento un método el cual crea una matriz de $n_x m$ la cual se fue llenando buscando la posición x,y de cada dato en el TDA matriz, esto evita que haya problemas al leer del xml si los datos de cada matriz vienen desordenados además de verificar que no tenga una posición fuera del rango de la matriz.

Se uso comparación de binarios para verificar la compatibilidad de las filas, estos binarios ayudan a simplificar que las filas se sumen correctamente, adicionalmente a esto evito la creación de una matriz binaria para su comparación.

Se opto por crear una copia de la matriz original y a esta aplicarle los métodos para su reducción por medio de los métodos binarios y de suma y eliminación de matrices, además en este proceso se determino la frecuencia de los grupos que conforman la matriz de datos reducida

Para la creación del xml se usó nuevamente la librería xml ElementTree, la cual nos facilita la escritura de este, además de poseer métodos específicos para la estructura correcta del archivo escrito de otra manera se escribiría como texto en el archivo, haciendo la comprensión de este demasiado difícil y poco ortodoxa.

En la creación del grafo de la matriz que se seleccione se utilizó la librería de graphviz la cual facilita la creación de nodos y de el correcto manejo de las ramas que se pueden crear, esto se implemento de manera simple pero legible.

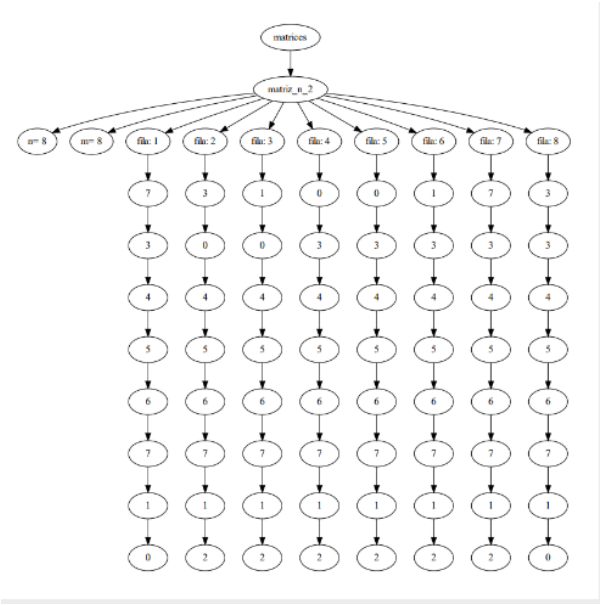

Figura 1. Grafo de matriz seleccionada.

Fuente: elaboración propia.

Se realizo el grafo de forma simple debido a la estructura en la que se trabajo cada lista y cada nodo, por lo cual se debe mencionar que se podría estructurar de mejor forma la matriz

Se opto por realizar la mayoría de los métodos en los TDA's lo cual facilito en gran medida la complejidad de iteraciones en los nodos de cada tipo que conforman las listas de matrices, filas, y datos. Esto evito que surgieran problemas de referenciación de nodos.

Se utilizo programación orientada a objetos para facilitar el almacenamiento de datos, y para simplificar los bloques de código.

Conclusiones

Se intento crear un programa simple, con una forma ligera para el procesamiento de datos procurando una correcta realización se ideo una metodología de programación orientada a objetos.

Se opto por realizar gran parte de los métodos en los TDA's, y se procuró realizar un código limpio en el cual se validaron las excepciones de forma simple y general.

Se evidencio que existen muchos problemas al no referenciar los nodos de forma correcta haciendo que estos sean un problema al realizar ciertos ciclos dentro de los métodos.

Se recomienda utilizar mensajes de traza frecuentemente para obtener referencias de ejecución de ciertos valores y verificar que lo que se este operando sea un valor y no una dirección de memoria.

Adicionalmente se recomienda que se busquen alternativas para cierto tipo de problemas siendo uno de estos un factor importante para la correcta verificación de funcionamiento de los módulos, siendo un claro ejemplo la correcta escritura de los archivos xml los cuales no se imprimen con una estructura legible sin utilizar métodos implementados en diferentes versiones de Python.

Se espera que se verifiquen primeramente sitios con documentación oficial y en el idioma de origen para obtener información detallada y precisa del funcionamiento de librerías y del lenguaje mismo antes de buscar en sitios externos tales como diferentes de los oficiales

Se recomienda realizar pruebas constantes para tener un punto de referencia para posibles fallas durante la construcción de la solución, esto evitara la complejidad de búsqueda de errores, además evitara el uso excesivo de herramientas como el debugger del IDE.

Se espera que el código sea implementado correctamente, también deberá estar correctamente versionado y comentado para no confundir diferentes segmentos de código que puedan ser similares.

Siendo la complejidad del problema un tanto elevada se busco siempre la forma más simple de solucionarlo.

Aunque siempre puede ser mejor y optimizado de mejor forma se implemento lo que se creyó en su momento los adecuado y lo necesario, pero siempre existe la posibilidad de hacerlo mejor.

Se recomienda hacer documentación que ayude al fácil desarrollo de la solución, tales como diagramas de flujo, diagramas de clase, diagramas de entidades, etc.

Se utilizo el software de control de vcrsiones GIT, el cual además se complemento con la integración de github en el IDE pycharm el cual facilito en gran medida la realización de commits los cuales tuvieron como objetivo mejorar el desarrollo de la solución.

Adicionalmente en github se manejo el control de los releases para la publicación de versiones funcionales del programa, esto también ayuda a la comunidad y el propio desarrollador o desarrolladores los cuales pueden tener retroalimentación de comentarios de los que han revisado el programa.

Referencias bibliográficas

Graphviz. (s. f.). *The DOT Language*. graphviz.org. Recuperado 8 de marzo de 2021, de https://www.graphviz.org/doc/info/lang.html

pypi. (2020, 24 diciembre). *Graphviz*. https://pypi.org/project/graphviz/#description

python.org. (s. f.). *xml.etree.ElementTree — The ElementTree XML API — Python 3.9.2 documentation*. python. Recuperado 8 de marzo de 2021, de https://docs.python.org/3/library/xml.etree.elementtree.html