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 – INTRODUCCION A LA PROGRAMACION Y COMPUTO 2

201700375 – Bryan Steve Montepeque Santos

Resumen

Se debe crear un Sistema de Bases de Datos alojado en varios sitios distribuidos para minimizar el costo de transmisión de datos, para esto se debe tener almacenada una matriz de acceso en los sitios de la instancia objetivo y transformarla en una matriz de patrones de acceso y agrupar las tuplas con los mismos patrones, es decir, debemos recibir los archivos con un archivo (En este caso un XML) y procesar todos los datos y almacenarlos en matrices, que al tener memoria limitada deben de ser implementadas con listas simples, es decir que cada matriz es una lista de filas donde cada una guarda una lista de datos para formar las columnas, esto se procesa para obtener una matriz binaria, ver las filas repetidas y sumarlas, esta matriz resultado es la matriz de patrones de acceso, y generar un archivo XML que almacene los datos de esta matriz de patrones.

Palabras clave

1. Lista Simple
2. Lista Circular
3. XML
4. Grafo Dirigido
5. Matriz Binaria

Abstract

We are required to create a Data Base System hosted on various distributed sites to minimize the data transmission cost, for this we have to have an Access Matrix store don the objective instance sites and transform it into an Access Pattern Matrix and add up the tuples with the same patters, in other words, we have to receive the data in a file (In this case an XML file) and process all the data and store them on matrixes, that since we have limited memory must be implemented with linked lists, in other words each matrix is a linked list, where each Row is a node that stores another linked list for the columns, this is processed to form a Binary Matrix, now we look for repeated rows in the matrix and add them up, the matrix that results from this is the Access Pattern Matrix, and then we generate and XML file that stores the data in this Pattern Matrix.

Keywords

1. *Linked List*
2. *Circular Linked List*
3. *XML*
4. *Directed Graph*
5. *Binary Matrix*

Introducción

El desarrollar este proyecto puede abrirles camino tanto a las empresas que no tengan tantos recursos como a las empresas que busquen reducir sus gastos el implementar un Sistema de Bases de Datos ya que este programa no necesitará de tanta memoria para transmitir los datos lo que reduce los gastos de equipo también, como ya fue discutido, que para este propósito implementaremos las matrices de la base de datos usando únicamente Listas Enlazadas ya que al ser estructuras dinámicas no se utiliza más memoria de la utilizada, pero ¿cómo serán implementadas? Esta es una pregunta respondida a continuación.

Desarrollo del tema

Para poder desarrollar este programa el proceso fue dividido en varios pasos:

1. Realizar el Menú:
   Este paso es bastante sencillo, solamente consiste en desplegar un conjunto de opciones y pedir que se ingrese una opción de todas las opciones que vé en la pantalla siendo estas:
   - Cargar el Archivo
   - Procesar el Archivo
   - Escribir Archivo Salida
   - Mostrar Datos del Estudiante
   - Generar Gráfica
   - Salir

   Para poder reconocer qué opción ingresó el usuario usaremos un input, y al leer su respuesta es comparada con un if else de varios casos para saber cuál de los métodos ejecutar, y este a su vez está encerrado por un try catch en caso de que el usuario decida ingresar un carácter invalido o lo haga por error

2. Cargar el Archivo XML:
   Para esto se le pide al usuario que ingrese la ruta absoluta del archivo incluyendo su extensión, si el archivo elegido no es un XML entonces muestra error y regresa al menú, pero si sí es un XML entonces utiliza la librería Element Tree para leer todas las etiquetas del archivo XML y las va identificando, luego de esto al identificar una Matriz almacena su nombre así como sus dimensiones y un apuntador al primer nodo de su lista de filas para poder acceder a la matriz más adelante, luego, al llegar a las etiquetas de datos, obtiene el valor de X y de Y que contiene este dato para poder ubicarlos dentro de nuestras listas, y comparamos el valor de X con el id de la Fila de la matriz ya que X representa el valor de la Fila donde va ubicado el dato entonces decimos que si el id de la fila no es igual a la fila que dice el dato entonces avance al siguiente nodo hasta que ambos sean iguales, una vez son iguales le decimos a la lista que guarda este nodo que agregue a el nuevo nodo en este caso, el dato que nos indica el archivo XML, después de repetir este procedimiento para todas las Matrices encontradas en el archivo XML muestra un mensaje al usuario donde se indica que todo el archivo ya se ha cargado, así como viene mostrando todo el procedimiento de almacenarlo con mensajes en la consola.

3. Procesar el Archivo:
   Aquí es en donde transformaremos la matriz en una Matriz de Patrones de Acceso, el primer paso para esto es hacer la matriz binaria a partir de la Matriz Ingresada, esto lo que hace es recorrer toda la matriz tanto en filas como en columnas e ir verificando por cada una “Si el valor de la matriz es diferente de 0 entonces el valor de la matriz = 1” y almacenar estos datos junto con su

valor original en otras matrices para luego operarlas, ahora debemos obtener una fila de la matriz en una lista simple auxiliar y compararla con una fila de la matriz y ver si su valor binario es idéntico, si sí es idéntico entonces se sumará el valor original de cada dato de esa Fila que estamos comparando con el valor original de la Fila de la matriz con la cual acabamos de comparar la lista auxiliar, y debemos repetir el proceso hasta que estén comparadas y sumadas todas las filas de la matriz binaria y la matriz resultado de este proceso es la Matriz de acceso, ahora bien, también debemos de crear una variable que lleve la cuenta de cuantas veces se repite cada Fila de la Matriz binaria para después mostrarla al usuario y también para guardarla en el Archivo XML que se generará luego Ahora debemos almacenar esta matriz de patrones de acceso así como la frecuencia de cada fila en otra matriz e informarle al usuario de esto mediante un mensaje en la consola, justo como durante todo este proceso se le debe informar al usuario acerca de qué parte del procedimiento está realizando en determinado momento.

4. Escribir Archivo Salida:
   Se debe escribir un archivo XML utilizando los datos de las Matrices de Patrones de Acceso, así como escribir la frecuencia de sus filas y el grado de cada matriz junto con sus dimensiones, para esto se debe recorrer la Matriz de Patrones de Acceso creada en el inciso anterior, así como ir concatenando un String que vaya conteniendo las etiquetas del XML y e ir recorriendo la lista y obteniendo sus datos al mismo tiempo, para esto se debe de recorrer la matriz tanto en filas como en columnas, normalmente se utilizarían dos fors anidados para esto, pero debido a la naturaleza de una Lista Enlazada, y de los Tipos de Datos Abstractos esto no es posible porque los nodos no son iterables, entonces debemos de utilizar un while dentro de otro while que vaya indicándole al nodo actual que después de concatenar un dato al string avance al siguiente nodo, esto debe de hacerse dentro de cada fila y al terminar de recorrer cada fila debe de repetir todo este proceso por cada una de las filas que tenga la matriz, para esto debemos de haber obtenido las dimensiones de la matriz antes de recorrerla ya que así tendremos cómo ir identificando cada posición de la matriz e indicarle al while cuando detenerse, después de recorrer toda la matriz ya solo debemos agregar las etiquetas de frecuencia para cada fila, y luego de esto decirle al programa que escriba este String en un archivo llamado “Archivo Salida.xml”.

5. Mostrar los Datos del Estudiante:
   Bien, solo se usan varios String para mostrar los datos del desarrollador de la aplicación, en este caso se mostrarán:
   - Nombre Completo
   - Carnet
   - Nombre Completo del Curso
   - Carrera del Estudiante
   - A qué semestre de dicha carrera pertenece el curso
6. Graficar las Matrices:
   En esta parte debemos de generar una gráfica en GraphViz de la Matriz que nos solicite el usuario, para esto solo debemos de desplegar todos las matrices almacenadas en nuestra matriz circular y pedirle al usuario que ingrese el número de la matriz que quiere ver graficada y con un if else comparar el valor que ingrese el usuario con el id de la matriz circular, si este id es igual al de cualquier matriz en la lista circular entonces ya procede a graficarla si no es igual al id de ninguna matriz en la lista circular entonces debe mostrarle al usuario que ingreso mal el dato o bien que no se encontró ninguna matriz con ese número, para esto debemos

utilizar un try catch, que si en dado caso el valor no es ninguno de los valores que se muestran tire una excepción, y vuelva a mostrar el menú y a pedir una opción hasta que el usuario ingrese una opción válida, después de esto, se procederá a graficar la matriz seleccionada para esto se utilizará un String que vaya concatenando el código de GraphViz (El cuál no verá el usuario, esto ya va escrito en el código del programa) y al mismo tiempo que este string se va concatenando también se va recorriendo la Matriz seleccionada por el usuario para poder obtener los datos correctos en cada fila y columna, así como también ir haciendo las conexiones correctas entre cada uno de los nodos de la matriz, junto con esta gráfica también se incluirán las dimensiones de Filas y Columnas de la matriz (llamadas "N" y "M" respectivamente) así como también mostrará el nombre de la matriz. Después de concatenar todo el String, así como en el inciso anterior debemos de escribirlo todo a un archivo llamado "Grafica Matriz.dot" y utilizaremos la librería Web Browser para usar su método Open para que automáticamente abra la gráfica de la matriz generada.

7. Salida:
   Cierra el programa usando sys.exit() que cierra el programa de Python y termina todos los procesos abiertos en el momento.

Conclusiones

El programa fue dividido en varios "Mini Programas" o módulos para poder simplificarnos la tarea de programarlo y también para poder repartirnos el trabajo entre varias personas si fuera necesario hacerlo así, esta dedicación y atención al detalle se ve reflejada a lo largo de todo el programa, empezando por el método del menú donde fue necesario usar una condición que verificara si una opción era válida y después usar un while que repitiera el ciclo try catch ya que sin el while el bloque try solo se ejecuta una vez, entonces fue necesario encerrarlo todo en un while y escribirle dentro del if else a cada opción válida que cambiara el valor de la variable de False a True para que el código del while dejara de repetirse, este nivel de calidad lo hemos puesto en todas las partes del programa así que creemos que utilizarlo será una experiencia agradable para nuestros usuarios.

Referencias bibliográficas

1. GeeksforGeeks. (2018, August 31). *Linked List Data Structure*. https://www.geeksforgeeks.org/data-structures/linked-list/

2. Pérez, G. M. C.-. (2014, July 15). *Grafo Dirigido | Grafos*. Http://163.10.22.82/OAS/Estructuras_de_grafos. http://163.10.22.82/OAS/estructuras_de_grafos/grafo_dirigido.html

3. *Matriz Lineal y Matriz Binaria*. (2014, June 22). WordPress.Com. https://mejorfuturoya.wordpress.com/matrizlinea-matrizbinaria/

4. GeeksforGeeks. (2018, September 7). *Circular Linked List | Set 1 (Introduction and Applications)*. https://www.geeksforgeeks.org/circular-linked-list/

5. *XML introduction - XML: Extensible Markup Language | MDN*. (2021, February 19). MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction