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.

EL PATRÓN DE ACCESO PARA UNA TUPLA

201700852 – Heli Saul Vásquez Gómez

Resumen

Este software es una aplicación en consola que posee varias funciones con el cual una de esas funciones es cargar archivos con extensión XML en forma de matriz el cual posterior mente es agregada en una lista de estructuras de datos para su realización y conversión de matriz identidad.

La búsqueda de patrones en base ala matriz identidad se basa en la suma de a filas similares a la demás y la muestra de frecuencias encontradas en las filas de la matriz introducida en la lista simple de estructuras de datos o TDA

Esta aplicación muestra la forma gráfica implementada con graphviz la matriz introducida dentro de una lista simple

Palabras clave

Palabra clave: Implementación de estructuras de datos TDA

Palabra clave: Conversión a identidad de matriz o matriz binaria

Palabra clave: Matriz de frecuencia de acceso

Palabra clave: implantación de la herramienta graphviz para la creación de grafos
Palabra clave: Cargar archivo con extensión XML

Abstract

In this software it is a console application that has several functions with which one of those functions is to load files with an XML extension in the form of a matrix which is later added to a list of data structures for its realization and conversion of the identity matrix.

The search for patterns based on the identity matrix is based on the sum of rows similar to the others and the sample of frequencies found in the rows of the matrix entered in the simple list of data structures or ADT

This application shows the graphical form implemented with graphviz the matrix entered within a simple list

Keywords

Keyword: Implementation of TDA data structures

Keyword: Conversion to Matrix Identity or Binary Matrix

Keyword: Access Frequency Matrix

Keyword: implementation of the graphviz tool for creating graphs

Keyword: Upload file with XML extension

Introducción

A continuación, en este presente documento o archivo realizaremos un análisis del uso de la aplicación de búsqueda de patrones que consiste en la conversión de una matriz a una matriz identidad o como una matriz en forma de binaria que posteriormente las filas similares son sumadas y agregadas en una lista simple o una TDA. Esta aplicación será realizada en el lenguaje de programación Python utilizando funciones necesarias para la correcta ejecución de la lógica de le programa o software. Posteriormente la escritura de salida del programa tendrá dos extensiones diferentes una la matriz resultante será escrita en un archivo XML con extensión XML la segunda salida del programa será en formato PNG para la visualización de un grafo de la matriz introducida con anterioridad en una lista simple. El software o programa contara con 6 opciones en un menú cada opción realizara las funciones mencionadas anteriormente como seleccionar un archivo o realizar conversión de la matriz.

Desarrollo del tema

Una clase es una Plantilla o modelo para crear a partir de ella ciertos Objetos. Esta plantilla es la que contiene la información; características y capacidades que tendrá el objeto que sea creado a partir de ella.

Así a su vez los objetos creados a partir de ella estarán agrupados en esa misma clase.

Una función es una porción o bloque de código reutilizable que se encarga de realizar una determinada tarea.

Teniendo definido que es una clase o función podemos continuar.

a. TDA
b. Implementación de clases
c. Implementación de funciones
d. Implementación de modulo Graphviz

TDA

Es un conjunto de datos u objetos al cual se le asocian operaciones. El TDA provee de una interfaz con la cual es posible realizar las operaciones permitidas

Implementación de clases

```
class dat1:
```

En esta clase se inicializaron las variables necesarias para la búsqueda de archivos y lecturas de las líneas del archivo.

```
class Nodo_1:
```

En esta clase se declararon las variables para posteriormente poder ser utilizadas en una lista doblemente enlazada

`class Nodo2_2:`
En esta clase se declararon las variables para posteriormente poder ser utilizadas en una lista simple

`class Nodo3_3:`
En esta clase se declararon las variables para posteriormente poder ser utilizadas en una lista simple en la cual esta lista se guardaron los patrones de búsqueda.

`class Lista:`
En esta clase creamos todas las funciones para la lista 1

`class Lista2:`
En esta clase creamos todas las funciones para la lista 2

`class Lista3:`
En esta clase creamos todas las funciones para la lista 3

Implementación de funciones Lista 1

`def Cargar ():`
La función cargar se ejecuta al momento de ser elegido la cual está encargada de abrir una venta y seleccionar un archivo. Esta función es complementada con el módulo Tkinter, esto nos ayuda a crear una interfaz grafica

`def Menú ():`
Esta función se crearon las funciones del menú para que el usuario pueda elegir con mas facilidad, las opciones que el desee, esta función se encargar generar un ciclo while en él se encuentran las opciones que se genera una tras otra

```
def Agregar (self, nombre, n, m):
```

Esta función pertenece a una lista que se encargar de agregar información por medio de parámetros que se cargar al momento de cargar el archivo.

```
def MostrarDatos(self):
```

Esta función se encarga de mostrar la información que son ingresados en la lista

```
def Modelo(self):
```

Esta función se encarga se simular la forma de una matriz, ya que en la lista cada información es agregada en un nodo

```
def ModeloBinario(self):
```

Esta función esta encargada de convertir nuestra matriz una matriz identidad o matriz binaria que luego nos fue útil para la búsqueda de patrones

```
def VerFormaMatrizBinaria(self):
```

En esta función nos nuestra la matriz binaria

Implementación de funciones Lista 2

```
def Agregar_2(self, x, y, numero):
```

Esta función perteneciente a la lista 2 agrega información de los sub datos del archivo XML , esta información es la posición X y Y el dato que corresponde a esas posiciones

```
def MostrarD(self):
```

Esta función nos nuestra los datos de la lista 2

```
def Simular_Matriz
(self, n, m):
```

Esta función se conecta las dos listas y toma como información los índices de los datos de la primera lista

Implementación de funciones Lista 3

```
def Agregar (self, contador, dato):
```

Esta función agrega los patrones buscados en la matriz binaria

```
def MostrarDatos(self):
```

Esta función muestra la información de la matriz reducida en base a los patrones

Implementación de modulo Graphviz

Es un módulo que nos sirve para La visualización de gráficos es una forma de representar información estructural como diagramas de gráficos y redes abstractos

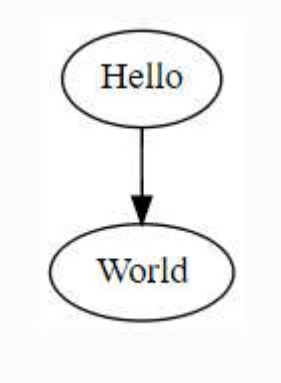
Ejemplo:

```
from graphviz import Digraph

g = Digraph ('G',
filename='hello. gv')

g. edge ('Hello', 'World')

g. view ()
```



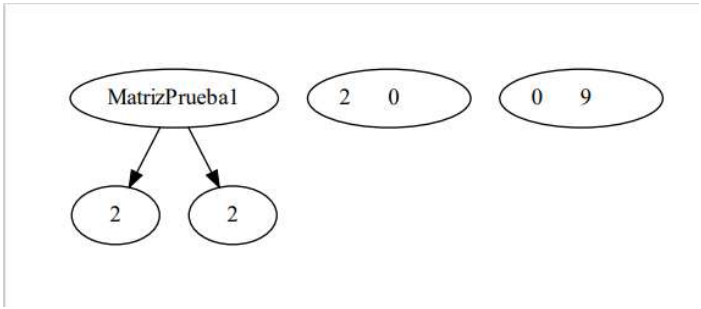
Fuente: elaboración propia, 2021 y 6.

Función Grafo

```
def e(self):
```

En esta función se crea el grafo o grafica de la matriz ingresa y el usuario elije cual matriz desea presentar en forma grafica

Ejemplo de la matriz ingresada:



Fuente: elaboración propia, 2021 y 6.

Esta imagen representa la matriz ingresa del archivo XML

Conclusiones

Esta aplicación representa la conexión entre listas simples y lista dobles enlazadas y el resultado de funciones para generar información grafica

Con la ayuda de las clases y las funciones se logro implementar la matriz binaria y la matriz reducida.

Las estructuras de datos implementadas en esta aplicación son una representación clara de lo mucho que se puede hacer con ellas

Referencias bibliográficas

Cairó y Guardati (2002)
ESTRUCTURA DE DATOS

JOYANES, L.; ZAHOHERO, I. (2005).
Programación en C: metodología, algoritmos y estructura de datos, 2ª Edición. Madrid: McGraw-Hill

Magnus Lie Hetland (2009)
Beginning Python, From Novice to Professional

Luis Joyanes Aguilar (1998)
ESTRUCTURA DE DATOS

Luis Joyanes Aguilar, Fernández Matilde, Rodríguez Luis (1999)
ESTRUCTURA DE DATOS Libro de Problemas