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.

ROBOT_ORTOGONAL – PROYECTO 2

201602659 – María Cecilia Cotzajay López

Resumen

Se implemento TDA´S específicamente Lista Enlazada Simple con su respectivo nodo, el cual contiene un apuntador y una Matriz Ortogonal.

El proyecto hace básicamente el procesamiento de Archivos con formato (.xml) el cual contiene una serie de datos para formar la lista matrices Ortogonales. Una vez hecha la carga de datos procede a graficar los nodos de cada matriz ortogonal por medio de la herramienta de Graphviz y se muestra en un documento (.pdf) que posteriormente se abre de forma automática.

Se realiza un análisis a través de una matriz previamente seleccionada. Finalmente se muestra un reporte en consola en base al resultado del análisis que se realizó sobre dicha matriz.

Palabras clave

i). **Nodo**: relación que incluye un puntero por cada lista así como información propia de la relación.

ii). **Matriz Ortogonal**: en este proyecto significa que son un tipo de lista que contiene nodos con cuatro apuntadores, lo cual lo convierte en una matriz, este es creado a mano, es decir sin librerías propias de Python.

iii) **Tipo de Dato Abstracto (TDA):** Modelo formal de un ente junto con un conjunto de operaciones definidas sobre el modelo que nos permite procesarlo.

.

Abstract

TDA's specifically implemented Simple Linked List with its respective node, which contains a pointer and an Orthogonal Matrix.

The project basically does the processing of files with format (.xml) which contains a series of data to form the Orthogonal matrix list. Once the data has been loaded, it proceeds to graph the nodes of each orthogonal matrix by means of the Graphviz tool and it is shown in the graphical part of the program

Different operations can be performed on an orthogonal matrix or between two of them. Finally, a detailed report is shown with the operations that were carried out on said matrices.

Keywords

*i). **Node**: relationship that includes a pointer for each list as well as information about the relationship.*

*ii). **Matrix**: in this project it means that they are simple lists within simple lists created by hand, that is, without Python's own libraries.*

*iii) **Type of Abstract Data (ADT)**: Formal model of an entity together with a set of operations defined on the model that allows us to process it.*

Introducción

Es muy importante aprender sobre el tema de los TDA´s así como su implementación en los diferentes proyectos. Cabe mencionar que la cualidad más importante sobre estos tipos de estructuras de datos es que son parte de la memoria dinámica, es decir que se puede utilizar para una cantidad grande de datos de los cuales muy probablemente desconozcamos el tamaño, es decir la cantidad. Si bien podemos mencionar otra cualidad de mayor importancia es que estos tipos de estructuras se pueden crear, manipular y manejar de la manera que nosotros queramos o gustemos.

El proyecto está basado en el uso, implementación y manejo de estas estructuras de datos, para tener un conocimiento profundo en el funcionamiento de las mismas, además de que nos dan un número finito de datos pero que al mismo tiempo es desconocido o no tentemos un número exacto de ellos.

En el siguiente apartado se describe lo que conlleva el proyecto, es decir cada parte del mismo.

Desarrollo del tema

El presente proyecto fue gráfico, es decir que el usuario interactúa con el programa mediante botones, ventanas, panel, etiquetas, barras de menú, ventanas de mensaje, ventanas que abren archivos, etc. Esto fue posible implementarlo en Python mediante la librería Tkinter. La forma en que trabaja esta librería no es con drag and drop, es decir que se hace cada elemento con código y se van editando la forma, el color de letra, el color de fondo, el tipo de letra, el tamaño, etc A mano.

Se realizó una ventana de presentación del proyecto, que redirecciona a la ventana principal del proyecto, el cual contiene una barra de menú, con las opciones de archivo que se subdivide en carga de archivos, reiniciar y salir. Esta también las Operaciones que se pueden realizar en una matriz o en dos matrices ortogonales (estas matrices ortogonales simulan ser imágenes tipo ascci), entre estas operaciones se encuentran rotación Horizontal, rotación vertical, transpuesta de Imagen o matriz, limpiar zona de imagen, (o eliminar nodos de matriz) agregar línea horizontal, agregar línea vertical, agregar rectángulo, agregar triangulo Rectángulo. En la siguiente opción

se puede encontrar el reporte, el cual es un archivo de formato (.html) el cual contiene el tipo de operación que se realizó sobre dicha o dichas imágenes, el nombre de la imagen, la hora y la fecha en que fuero operadas. La siguiente opción es la de Ayuda que contiene los datos de mi persona (en una ventana por aparte), que fue quien elaboró el proyecto, así como la opción de documentación que al presionarlo abre directamente el presente documento. En la interfaz también se puede apreciar la etiqueta con el nombre del curso y 3 etiquetas que contienen imágenes quemadas.

En la parte de código, el presente proyecto utilizo una clase llamada Presentación que es la que contiene toda la parte gráfica del proyecto, a partir de allí, se creó una clase llamada Metodos que es la que lleva la estructura del código, es decir los métodos principales, tales como, cargar Archivo, el cual abre un archivo mediante la ruta del archivo que usuario ingresara en la consola, hace la lectura correspondiente, extrae y ordena los datos de manera conveniente para posteriormente almacenarlos en las respectivas TDA´s. El siguiente método es graficar matriz, este método únicamente grafica los datos de cada nodo de la matriz ortogonal en consola. El método mostrar Documentación que como bien se ha mencionado antes, lo que hace es abrir el presente documento. El método Reporte HTML, llama a otros métodos que se encuentran en la presente Clase, que son crear Archivo, el cual genera un archivo con almacenamiento en el escritorio de nombre 'Reporte' el otro método llamado escribir línea, lo cual está compuesto por una cadena que se subdivide en otras tres cadenas, la primera contiene la parte inicial de la estructura de un archivo (.html) la segunda que es una concatenación del cuerpo del documento html más la línea que se quiere agregar, por último el tercero contiene la finalización de la estructura html, para posteriormente escribirlo en el archivo como una sola cadena de texto.

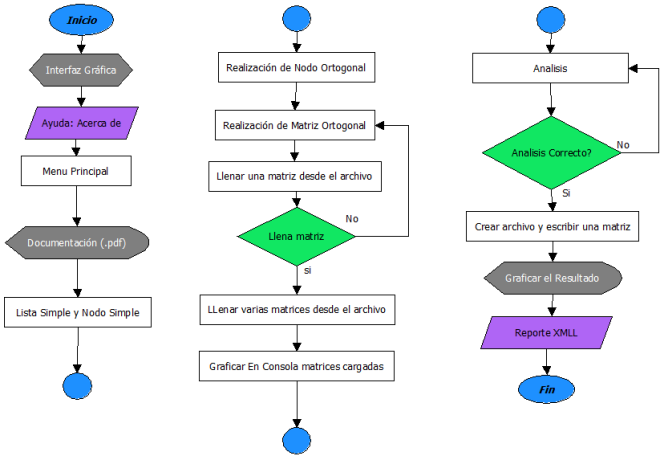
Como bien se ha mencionado antes se hizo la implementación de una lista Simple y su respectivo nodo Simple, el cual contiene un apuntador llamado siguiente, el nombre de la matriz y la matriz ortogonal. Los métodos de la lista simple son insertar, buscar nombres (que se usa para verificar si la existencia de una matriz con el mismo nombre, retorna Verdadero si encuentra una matriz con dicho nombre).

Para poder realizar la matriz ortogonal fue necesario el uso de una clase llamada Lista Ortogonal el cual hizo uso de una clase llamada Nodo Ortogonal Matriz, el cual contiene un objeto llamado Casilla y cuatro apuntadores: siguiente, anterior, arriba y abajo respectivamente. La clase Casilla contiene únicamente los atributos posicionX, posicionY y el dato o valor. La clase Lista Ortogonal contiene los siguientes métodos: insertar cabecerasY, lo cual hace nodos de tipo cabecera y las inserta en el eje de las Y y 0 en el valor X; el método insertar cabecerasX el cual hace el mismo procedimiento que el método anterior solo que con 0 en Y y las inserta en el eje de las X o abscisas; el método obtener cabeceraY el cual recibe como parámetro el valorY de una casilla, para obtener la cabecera de dicho valor Y; el método obtener cabeceraX hace exactamente lo mismo que el método anterior solo que con X.

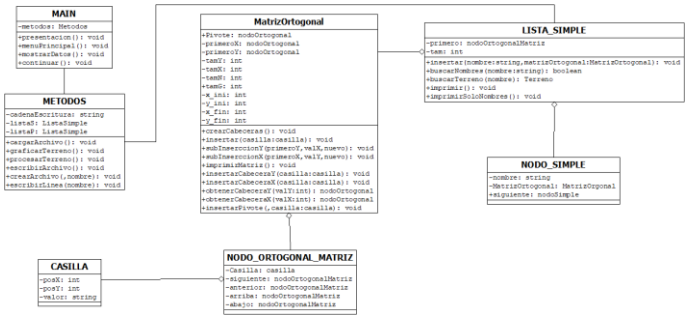
Por último se encuentra la clase llamada Matriz ortogonal que recibe como parámetros el tamaño de las filas y el de las columnas, utiliza un objeto de tipo Lista Ortogonal. Esta clase tiene métodos como crear cabecera, el cual crea las cabeceras en su respectivo eje, dependiendo del número de filas y columnas que se toman de los parámetros del constructor de la clase; insertar, que lo que hace es crear objetos de tipo casillas y los inserta en el objeto tipo Lista Ortogonal, utilizando los apuntadores necesarios (en algunos nodos se utilizan los cuatro), por último se encuentra el método imprimir matriz

A continuación se presenta una imagen del diagrama de flujo que se llevó a cabo para realizar la estructura general del proyecto:

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.

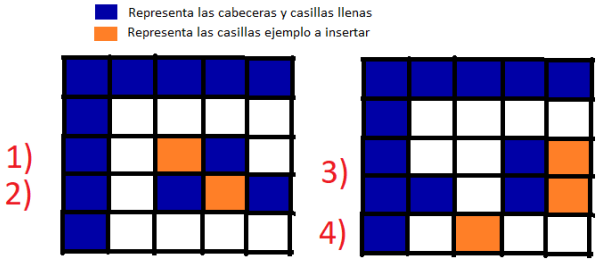


El siguiente diagrama de es el de Clase, muestra las clases que se utilizaron para llevar a cabo este proyecto:



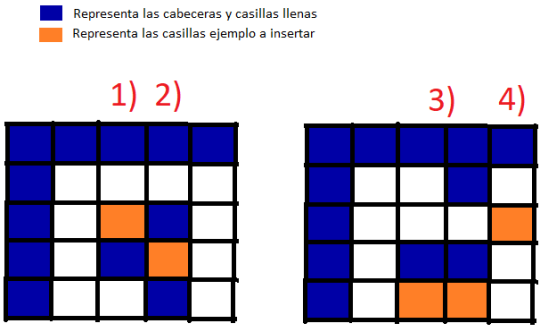
Imágenes de apoyo:

Inserccion en Y

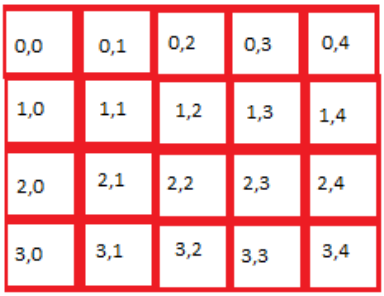


Inserccion en X:

el



Otra imagen de apoyo:



Conclusiones

Efectivamente los TDA´s son implementadas a gusto de cada quien, pues hay formas diferentes de hacerlas.

El uso de los TDA´s en el proyecto hizo que comprendiera funcionamiento en si de una matriz y más que eso, ya que al ser Ortogonales o también llamadas matrices dispersas, los datos estarían como flotando, eso depende mucho del número de apuntadores que contiene cada nodo. El número de apuntadores es lo que vuelve a un nodo disperso y por tanto una matriz dispersa.

El procedimiento de la carga de Archivos hizo comprender que el procedimiento con los TDA´s es muy largo pero que a su vez es fácil encontrar posibles errores ya que al crearlas uno mismo es posible encontrarlos muy fácil y rápidamente

Referencias bibliográficas

Máximo 5 referencias en orden alfabético.

C. J. Date, (1991). *An introduction to Database Systems.* Addison-Wesley Publishing Company, Inc.

MULTILISTAS (ugr.es) (Para consulta de algunas palabras clave.)