
DESARROLLO DE LA APLICACIÓN PARA REALIZAR OPERACIONES CON MATRICES ORTOGONALES.

200011006 – Adolfo Francisco López Cuzco

Resumen

En la actualidad el uso de matrices ortogonales en diferentes sitios es extensa y por consiguiente sugiere que es la mejor manera para el manejo de memoria ya que para el procesamiento de toda la información y asegurar que los recursos se aprovechen a lo máximo y no se desperdicien guardando datos que no se necesitan. Estas estructuras se pueden utilizar en las para el mejoramiento de la cantidad de información en la web.

Para resolver el problema se desarrolla una aplicación utilizando un TDA o tipo de dato abstracto, este tipo de dato es una Matriz ortogonal la cual se debe crear con una cabecera de columnas y otra cabecera de filas además se deben enlazarlas a una lista doblemente enlazada.

Palabras clave

TDA
Matriz Ortogonal
Nodo
Cabecera

Abstract

At present, the use of orthogonal matrices in different places is extensive and therefore suggests that it is the best way for memory management since for the processing of all the information and ensuring that the resources are used to the maximum and are not wasted. saving data that is not needed. These structures can be used in companies to improve the amount of information on the web.

To solve the problem, an application is developed using an ADT or type of abstract data, this type of data is an orthogonal matrix which must be created with a column header and another row header, and they must also be linked to a doubly linked list.

Keywords

TDA
Orthogonal Matrix
Node
Headboard

Introduction

Para el manejo de operaciones con matrices es necesario realizar una investigación sobre las maneras en que se puede crear la matriz ortogonal. Esto será de gran ayuda porque al utilizar el método más óptimo nuestra aplicación tendrá la facilidad del manejo del almacenamiento de datos por medio de dicha matriz.

Para el estudio se debe conocer sobre la forma en que están estructuradas, modeladas o diseñadas las matrices ortogonales.

Existen modelos que se pueden aplicar para encontrar un patrón de uso de matrices ortogonales.

Para el desarrollo de la aplicación se utilizará una matriz ortogonal y luego representar las operaciones dicha matriz dentro de una interfaz gráfica.

Desarrollo del tema

Para poder desarrollar la aplicación se utilizó la herramienta de desarrollo Python 3.8. y el editor Atom. Existen buenas razones técnicas para utilizar estas nuevas herramientas de desarrollo ya que brindan un gran espectro en cuanto a las librerías que posee. Para este proyecto se decidió utilizar una librería denominada tkinter de Python 3.8 para poder realizar la interfaz gráfica de la aplicación.

Para el desarrollo de la solución del problema es necesario saber cómo funciona el procesamiento de

estructuras de datos dentro de ellas esta la matriz ortogonal y el manejo de ellas para poder realizar operaciones sobre ellas

a. TDA

Un Tipo de dato abstracto (en adelante 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, abstrayéndose de la manera en cómo estén implementadas dichas operaciones. Esto quiere decir que un mismo TDA puede ser implementado utilizando distintas estructuras de datos y proveer la misma funcionalidad.

El paradigma de orientación a objetos permite el encapsulamiento de los datos y las operaciones mediante la definición de clases e interfaces, lo cual permite ocultar la manera en cómo ha sido implementado el TDA y solo permite el acceso a los datos a través de las operaciones provistas por la interfaz.

En este capítulo se estudiarán TDA básicos como lo son las listas, pilas y colas, y se mostrarán algunos usos prácticos de estos TDA.

e. TDA lista

Una lista se define como una serie de N elementos E_1, E_2, \dots, E_N , ordenados de manera consecutiva, es decir, el elemento E_k (que se denomina elemento k -ésimo) es previo al

elemento E_{k+1} . Si la lista contiene 0 elementos se denomina como lista vacía.

Las operaciones que se pueden realizar en la lista son: insertar un elemento en la posición k , borrar el k -ésimo elemento, buscar un elemento dentro de la lista y preguntar si la lista está vacía.

f. Lista circular simplemente enlazada:

No existe ningún nodo que apunte a null. La lista no tiene fin ya que al llegar al último nodo empieza de nuevo la lista. Se accede a la lista mediante el primer nodo o también llamado inicio de la lista

g. Matriz Ortogonal

Los nodos en esta estructura de datos suelen tener varios apuntadores.

Esta formada por un conjunto de listas doblemente enlazadas.

Cada lista tiene una cabecera, por medio de la cual se pueden acceder a la matriz.

La cabeza principal apunta al primer cabeza de filas y al primer cabeza de las columnas.

Para Realizar el proceso de inserción, se siguen los siguientes pasos.

1. Se recorren los encabezados, los cuales deben encontrarse ordenados, para verificar si existe el encabezado necesario, si este no existe se crea.

2. Si el encabezado ya hubiera existido, quiere decir que ya hay elementos en esas dimensiones. Entonces se inserta de forma ordenada en esa lista el elemento nuevo.
3. Se recorre los encabezados de la siguiente dimensión, buscando el necesario, si no existe se crea.
4. Si el encabezado de la siguiente dimensión ya hubiera existido, se asocia en la lista el nuevo elemento

Solución del Problema

Tomando en cuenta lo investigado se procede a elaborar los algoritmos necesarios para crear la matriz ortogonal.

Se crea un nodo con los siguientes atributos.

fila=nfila

columna=ncolumna

dato=dato

izquierda=None

derecha=None

arriba=None

abajo=None

Se crear una lista de cabeceras para poder enlazar los elementos de la matriz y relacionarlos con dichas cabeceras.

Su atributo principal es el

Inicio.

Continente los métodos

CrearCabecera.

DevolverCabecera.

También se crea un nodoCabecera que contiene atributos siguientes.

```
self.codigo=codigo
```

```
self.cbrAnterior=None
```

```
self.cbrSiguiente=None
```

```
self.nodoDato=None
```

Este nodo es para manipular las cabeceras.

Luego procedemos a crear el modulo matriz en este módulo se creará una clase Matriz.

Con los siguientes atributos y métodos.

```
self.nombre=nombre
```

```
self.cabFila=ListaCabecera()
```

```
self.cabColumna=ListaCabecera()
```

```
self.dato=None
```

```
self.tamano=0
```

Métodos Básicos:

Agregar()

Modificar()

Eliminar()

Desplegar()

Buscar()

Tambien agregaremos métodos para poder realizar las diferentes operaciones ya sea con una matriz o con dos matrices.

Dentro de estos métodos podemos listar los siguientes.

llenarMatriz()

girarMatrizHorizontal(matriz)

girarMatrizVertical(matriz)

transpuestaMatriz(matriz)

agregarLineaHorizontal(fila,columna,elementos)

agregarLineaVertical(fila,columna,elementos)

agregarRectangulo(fila,columna,elementos)

agregarTriangulo(fila,columna,elementos)

También debemos crear un modulo para poder cargar los archivos a memoria para ello debemos crear la clase Archivo.

Que posee los siguientes métodos.

Abrir(rutaDelArchivoXml)

graficarMatrices(matriz)

crearArchivosHtml()

Para la interfaz gráfica o capa de presentación se debe crear

En esta clase debemos agregar el menú para la manipulación de matrices.

Para ello crearemos una clase Principal en la cual agregaremos el menú, los botones para poder realizar las operaciones con una y con dos matrices además debemos agregar la opción para genera los reportes HTML. Y por ultimo los datos del estudiante.

Los métodos principales para esta clase son.

cargarArchivo()

graficarMatriz()

OperaciónConUnaMatriz()

OperaciónConDosMatrices.()

Reportes()

Ayuda()

Conclusiones

El manejo una matriz ortogonal es de gran importancia dentro de la estructura de datos y el manejo de memoria dinámica ya que asignando los valores dentro de una matriz ortogonal ahorramos espacio en memoria lo cual acelera el procesamiento de datos.

Para optimizar y reducir el consumo en memoria se aconseja que mejor utilizar una matriz ortogonal elaborada con dos listas doblemente enlazada y con sus cabeceras respectivas. De otro modo si utilizamos una lista no se tiene control en el manejo de los datos de la matriz.

Referencias bibliográficas

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