
PROYECTO2 DE IPC2

201900716 –Kelly Mischel Herrera Espino

Resumen

En el presente ensayo se explica la implementación de una estructura de datos en este caso una matriz dispersa, la cual cuenta con un encabezado en las filas y uno en las columnas. La implementación de esta EDD es complicada y el proceso es tardado, pero se pueden guardar los datos de mejor manera, así como obtenerlos. En este proyecto se utilizó una matriz ortogonal para almacenar los datos en este caso se puede almacenar más de una matriz de n tamaño se cuadrado o no. También se implementó una lista simple en la cual se guardaban las matrices ortogonales. Al tener los datos almacenados se podían aplicar varias operaciones a una sola matriz como el cambiar de forma horizontal la forma original intercambiando las filas de lugar la primera quedando en la base y la última en la primera línea, parecido a esto fue el cambiar de posición de forma vertical los datos de la matriz.

Palabras clave

- EDD: Estructura de Datos
- Matriz: Es un arreglo bidimensional de números
- Implementación: Es incorporar al código
- Lista: Es una secuencia de datos ordenados
- Matriz Ortogonal: Es aquella cuyos nodos suelen tener varios apuntadores

Abstract

This essay explains the implementation of a data structure in this case a sparse matrix, which has a header in the rows and one in the columns. The implementation of this EDD is complicated and the process is time consuming, but the data can be saved in a better way, as well as obtained. In this project a sparse matrix is used to store the data, in this case more than one matrix of n size can be stored, whether squared or not. A simple list was also implemented in which the orthogonal matrices were stored. By having the data stored, several operations can be applied to a single matrix, such as changing the original shape horizontally, exchanging the rows of place, the first remaining in the base and the last in the first line, similar to this was changing position the matrix data vertically.

Keywords

- *Matrix: It is a two-dimensional arrangement of numbers*
- *Implementation: It is to incorporate the code*
- *List: It is a sequence of ordered data*
- *Orthogonal Matrix: It is one whose nodes usually have several pointers*

Introducción

En el presente ensayo se dan a conocer algunas características de una matriz ortogonal y a su vez se explica el porque es de gran ayuda el implementar una estructura como esta para almacenar datos, sin perder la referencia y siendo esta sencilla de manejar. Para este proyecto se buscaba realizar las distintas operaciones ya sea de una matriz o entre dos matrices recorriendo los datos de las listas ortogonales almacenadas en un matriz simple. Por medio del encabezado se recorrían los dato y por medio de un id se le aplicaba la operación a dicha matriz. Las operaciones se mostraron en una interfaz gráfica creada por medio de la librería tkinter y con ayuda de graphviz se creaban imágenes con tablas las cuales contenían lo cambios que se le realizaba a cada matriz en especifico. Al usar esto se obtenía la imagen y se colocaba en el frame.

Debe contener un máximo de 150 palabras.

Desarrollo del tema

Antes de implementar la matriz ortogonal se debe definir algunas características que esta tiene:

Características:

- Los nodos en esta estructura suelen tener varios apuntadores

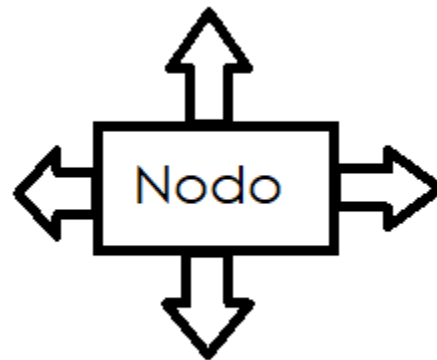


Figura 1. Ejemplo de los apuntadores de un nodo de un matriz Ortogonal.

Fuente: elaboración propia, Kelly Herrera, 05 de abril 2021

- Esta formada por un conjunto d elista doblemente enlazadas.

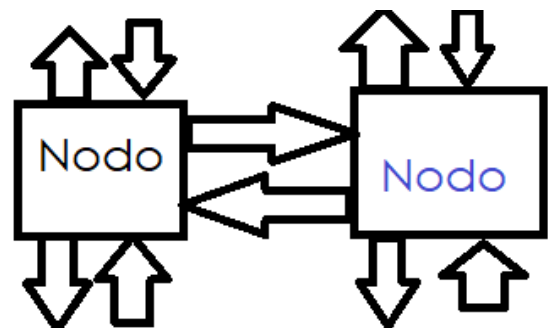


Figura 2. Ejemplo de los apuntadores de una lista doblemente enlazada.

Fuente: elaboración propia, Kelly Herrera, 05 de abril 2021

- Cada lista tiene una cabecera por medio de la cual se puede acceder a la matriz.
- La cabecera principal apunta a la primer cabeza de filas y a la primer cabeza de columnas.

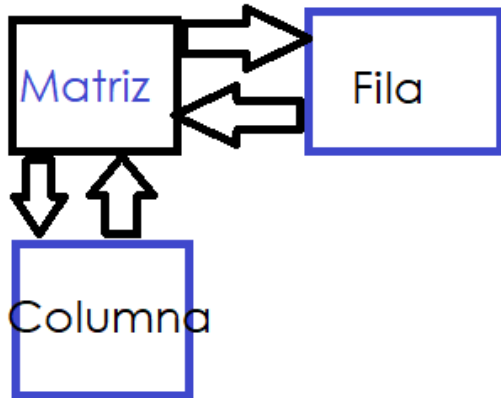


Figura 3. Ejemplo de como apunta la cabeza a las primer cabeza columna y fila.

Fuente: elaboración propia, Kelly Herrera, 05 de abril 2021

a. Implementación de una matriz Ortogonal

Para implementar una matriz ortogonal se crea una clase Nodo esta contiene los valores que van dentro de los nodos que apuntan a los encabezados de fila y columna.

Luego se crea una clase nodoEncabezado la cual contendrá los nodos encabezados con su respectivo valor. También se crea una clase encabezado en donde se valida el orden que tendrá tanto el encabezado de las filas como el de las columnas. Por medio de ifs se compara el id único que le corresponde a cada nodoEncabezado para guardarlo de menor a mayor recorriendo esto con un while mientras cumpla con la condición.

Para crear la matriz se utiliza una clase en este caso se llama de la misma forma, en esta parte se hace la mayor parte de validaciones para poder almacenar los datos y que los apuntadores queden como se muestra en la figura 3.

b. Operaciones para una sola matriz.

Hay 8 operaciones de las que el usuario tiene a disposición eligiendo a conveniencia cualquier de ellas.

Las operaciones son las siguientes:

1. Rotación horizontal de una imagen

En esta operación se recorrió la matriz con un for en un rango de mayor a menor. De esta forma se pudo obtener el resultado deseado el cual era que las filas cambiaran de posición siendo la primera la última y así sucesivamente.

```
for i in range(fila-1, -1, -1):  
    # print("hoed")  
    # print(i)  
    print("valor ")  
    print(temp.matriz.retornarFila(i))  
    cadena += str(temp.matriz.retornarFila(i))
```

Figura 4. Fragmento de código de como se realizó el cambio posición horizontal.

Fuente: elaboración propia, Kelly Herrera, 05 de abril 2021

2. Rotación vertical de una imagen

Esta operación se realizó parecida a la anterior solo que en este caso se hizo el cambio de posición entre columnas para ello también se utilizó un for y un método que retornara el valor que contenía cada columna para colocarlo en su posición contraria.

```
for i in range(columna-1, -1, -1):  
    # print("hoed")  
    # print(i)  
    print("valor ")  
    print(temp.matriz.retornarColumna(i))  
    cadena += str(temp.matriz.retornarColumna(i))
```

Figura 5. Fragmento de código de como se realizó el cambio posición vertical.

Fuente: elaboración propia, Kelly Herrera, 05 de abril 2021

3. Transpuesta de una imagen

En este caso se recorrió la matriz con dos fors en el primero se colocó en el rango la cantidad de columnas y en el segundo se colocó la cantidad de filas. Esto con el fin de que las filas quedaran en la posición de las columnas y viceversa.

```
for i in range(columna):  
    for j in range(fila):  
        # matriz.insertar(j, i, cadena[c])  
        print("\t", cadena[c], end=" ")  
        matriz.insertar(i, j, cadena[c])
```

Figura 5. Fragmento de código de como se realizó la transpuesta.

Fuente: elaboración propia, Kelly Herrera, 05 de abril 2021

4. Limpiar zona de una imagen

En esta operación se limpiaba una zona según el usuario ingresar la posición en la fila y columna, así como la cantidad de elemento que iba a borrar.

5. Agregar línea horizontal a una imagen

En esta operación se agregaba una línea horizontal de los de asteriscos para ello se colocaba la fila y luego la cantidad de elementos que se iban a asignar.

6. Agregar línea vertical a una imagen

Esta operación es parecida a la anterior solo que en este caso se agregaba una línea vertical de los de asteriscos para ello se colocaba la columna y luego la cantidad de elementos que se iban a asignar

7. Agregar rectángulo

En esta operación se colocaba un rectángulo de asteriscos para ello se necesitaban los datos de la posición en la fila y en la columna, y también el ancho y la altura que tendría este rectángulo.

8. Agregar triángulo rectángulo

En esta operación se agrega a la matriz un triángulo rectángulo.

Conclusiones

La implementación de una estructura de datos en este caso la matriz dispersa suele verse complicada de implementar y de manejar los datos. Sin embargo cabe mencionar que el uso de esta matriz ayuda al manejo y almacenamiento de los datos sin mayor dificultad.

Se puede crear una matriz ortogonal son con el encabezado de las filas, sin afectar su funcionamiento aunque en este proyecto se trabajó con ambos encabezados para un mejor manejo de los datos y la referencia de los nodos no se perdiera.

Referencias bibliográficas

Máximo 5 referencias en orden alfabético.

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

