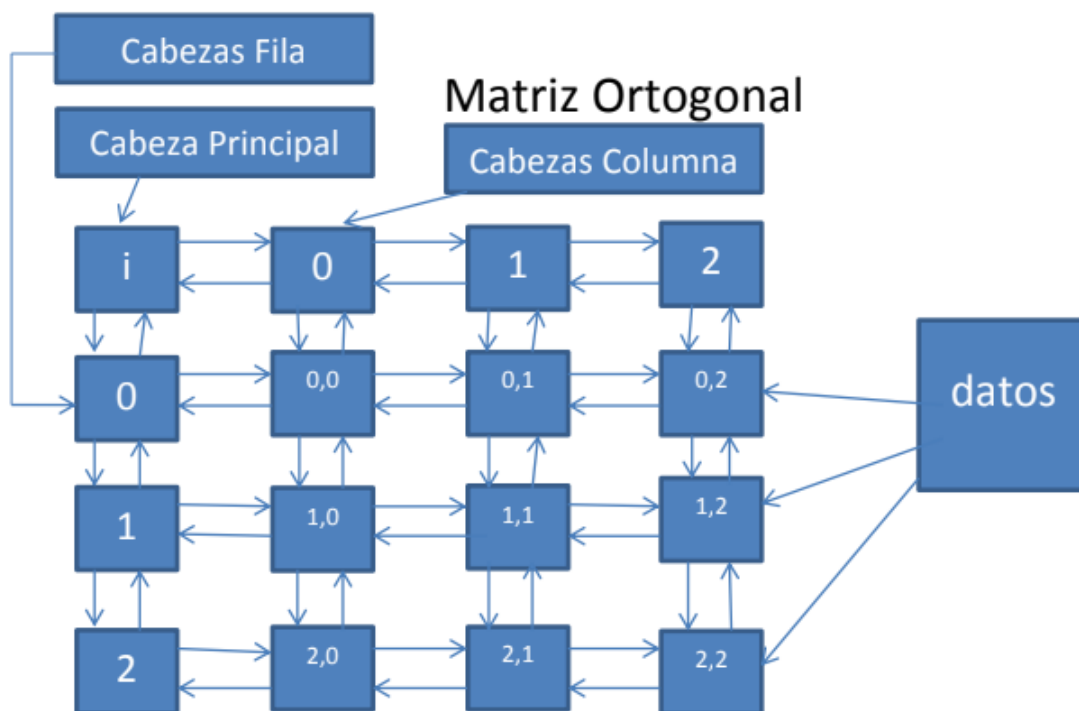


Manual Técnico

Estructura de datos

Una estructura de datos es una forma particular de organizar datos en una computadora para que pueda ser utilizado de manera eficiente. Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas. Las estructuras de datos son un medio para manejar grandes cantidades de datos de manera eficiente para usos tales como grandes bases de datos y servicios de indización de Internet. Por lo general, las estructuras de datos eficientes son la clave para diseñar algoritmos eficientes. Algunos métodos formales de diseño y lenguajes de programación destacan las estructuras de datos, en lugar de los algoritmos, como el factor clave de organización en el diseño de software. Las estructuras de datos se basan generalmente en la capacidad de un ordenador para recuperar y almacenar datos en cualquier lugar de su memoria.

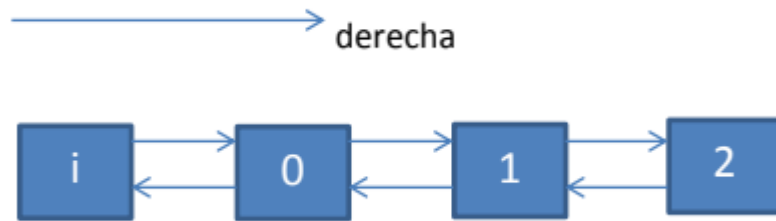
Tablero realizado por matriz ortogonal



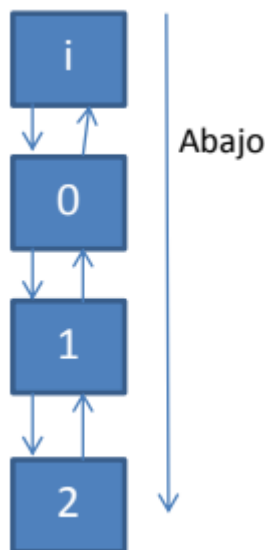
Una matriz ortogonal se compone por cuatro punteros “Arriba”, “Abajo”, “Derecha”, “Izquierda” básicamente la matriz ortogonal es un conjunto de listas doblemente ligadas donde cada lista tiene una cabeza ya sea en forma vertical “0,1,2” o horizontal “0,1,2” y a su vez estas cabezas tienen una cabeza principal, en el caso de la figura 1 está representado como “i”

Tomando como base el nodo “i” de la figura 1 vamos decir que todos los nodos que están a la derecha del Nodo “i” son Nodos Columna es decir representan las cabezas de columnas de la matriz.

Tomando como base el nodo “i” de la figura 1 vamos decir que todos los nodos que están a la izquierda del Nodo “i” son Nodos Fila es decir representan las cabezas de filas de la matriz.



Tomando como base el Nodo “i” de la figura 1 vamos a decir que todos los Nodos que están hacia abajo del Nodo “i” son Nodos Fila es decir representan las cabezas de Filas de la matriz.



Para poder hacer la inserción de la matriz ortogonal se van utilizar dos métodos los cuales son: NuevaColumna() , NuevaFila() y las siguientes restricciones

- La cabeza Principal ya deber existir para para poder hacer las inserciones.
- Las cabezas de fila y columna no de ven aparecer ocupando memoria si no hay un dato específico en esa posición por ejemplo: si esta lista de coordenadas (0,2) (1,2) (2,2) (2,0) (2,1) no existiera entonces eso significa que la cabeza 2 de columna y cabeza 2 de Fila no existen por el momento.

- Las inserción se divide en dos sub inserciones las cuales son:
Inserción de cabezas (columna y fila)

- o Insertar Columna

- Se tiene que utilizar el método NuevaColumna()

- o Insertar Fila

- Se tiene que utilizar el método NuevaFila()

- Inserción de datos (Columna y fila)

Dado la naturaleza de los métodos para insertar Columna y Fila de las cabezas (Columna, fila), se puede re-utilizar los mismos métodos para poder ingresar datos, solamente intercambiando el orden, a simple vista se mira confuso dado a que para insertar un dato en la cabeza de la columna se tiene que utilizar el método NuevaFila(), porque este método ingresa Nodos hacia abajo ajustándose a la necesidad de la cabeza Columna, en cambio el método NuevaColumna() ingresa Nodos hacia la derecha ajustándose a la necesidad de la cabeza Fila.

o Insertar Dato en Columna

Metodo a utilizar NuevaFila()

o Insertar Dato en Fila

Metodo a utilizar NuevaColumna()

Buscar Nodo (NodoCen)

Se puede buscar en los siguientes órdenes les da el mismo resultado

- Primero Columna y luego Fila
- Primero Fila y luego Columna

Tomando como referencia Nodo “i” de la figura 1 se busca hacia la derecha para poder encontrar la columna correspondiente y luego se busca hacia abajo para poder encontrar el nodo buscado si el nodo no existe se manda un mensaje de inexistencia.

Eliminar nodo

Para poder eliminar un Nodo solamente se tiene que hacer un sondeo de los cuatro apuntadores que tiene el nodo a las diferentes localidades de memoria re-apuntar los nodos.

Lista simple dinámica

Una lista es una estructura de datos que nos permite agrupar elementos de una manera organizada. Las listas al igual que los algoritmos son importantísimas en la computación y críticas en muchos programas informáticos.

Las listas están compuestas por nodos, estos nodos tienen un dato o valor y un puntero a otro(s) nodo(s).

Existen varios tipos de listas: Simplemente enlazada, doblemente enlazada, circular simplemente enlazada, circular doblemente enlazada.

Vamos a revisar las listas enlazadas simples, por ser el punto de partida y fundamentales para poder entender las otras.

Una lista enlazada tiene un conjunto de nodos, los cuales almacenan 2 tipos de información: El dato que contienen y un puntero al siguiente nodo en la lista. El último nodo de la lista tiene como siguiente nodo el valor NULL. Entonces las listas enlazadas simples solo pueden ser recorridas en una dirección, apuntando al nodo siguiente, mas no a un nodo anterior.

Métodos que utilizamos

- Añadir un elemento al inicio.
- Añadir un elemento al final
- Añadir un elemento de manera ordenada
- Imprimir la lista
- Buscar un elemento

Diagrama

