Escuela de Ingeniería Informática - Guayana

Algoritmos y Programación II

Proyecto 1

Una matriz dispersa o esparcida, es aquella que contiene relativamente pocos elementos no nulos, es decir la mayoría de sus elementos son ceros. En la representación de matrices con estructuras secuenciales, utilizando arreglos (un arreglo para cada fila y que el número de elementos están determinados por el número de columnas) puede derrochar bastante memoria si estamos tratando con Matrices Dispersas.

Se puede realizar un planteamiento alternativo utilizando estructuras enlazadas (listas) para almacenar los elementos no nulos de la matriz, por ejemplo, almacenar todos los elementos de una matriz de 1.000 x 1.000 requeriría de asignar memoria para un 1.000.000 de elementos. Sin embargo, si esta matriz sólo contuviera de 20.000 elementos no nulos y se emplease el planteamiento con lista solamente se almacenarían 20.000 elementos. Comparada con la implementación con arreglos ahora se requeriría más trabajo para acceder a un elemento en particular, pero se ahorra tiempo al llevar a cabo operaciones sobre matrices dispersas, dado que se considera los elementos no nulos, de este un elevado número de operaciones aritméticas inútiles (p.e. sumas de cero y multiplicaciones por cero) pueden ser evitadas.

Existen múltiples implementaciones, por ejemplo, se pueden utilizar arreglos siendo el caso del Formato Compressed Row Storage (CRS) o el Formato Compressed Column Storage (CCS). Pero es de interés en este proyecto el uso de memoria dinámica y realizar una implementación utilizando listas simplemente enlazadas, es decir para representar una matriz se construye una lista que almacene el identificador de la fila junto con el vector asociado representado como una lista, para la matriz identidad de 4x4 se tiene: ((1,((1,1))),(2,((2,1))),(3,((3,1))),(4,((4,1)))). Al ser una estructura donde el acceso por columnas no es eficiente se debe estudiar detalladamente el caso de la multiplicación de matrices y como parte del proyecto debe presentar una solución.

Para efecto del proyecto, una matriz se define como una organización

rectangular de elementos organizados en n filas y m columnas, y debe implementar las siguientes funciones:

ObtenerElemento(i,j,M). Devuelve el elemento de la fila i y la columna j de la matriz M.

AsignarElemento(i,j,x,M). Asigna x al elemento de la fila i y la columna j de la matriz M.

Sumar(M_1,M_2). Devuelve la matriz resultante de sumar M_1 y M_2, las matrices M_1 y M_2 deben tener la misma dimensión.

ProductoPorEscalar(e,M). Devuelve la matriz resultante de multiplicar M1 por el escalar e.

Producto(M_1,M_2). Devuelve la matriz resultante de multiplicar M_1 y M_2, el número de columnas de M_1 debe ser igual al número de filas de M_2. La matriz resultante tiene el mismo número de filas de M_1 y el mismo número de columnas de M_2.

Transponer(M). Devuelve la transpuesta de M.

Imprimir(M). Muestra la matriz M.

Puede definir otras funciones por ejemplo leer o escribir una matriz en almacenamiento secundario, así mismo en las estructuras de datos puede incluir campos que faciliten las diferentes funciones, justificando los mismos. La implementación debe realizarse en Lenguaje C, utilizando compilación por separado (matriz.h, matriz.c y proy1.c) y solo está permitido el uso de las librerías estándar del lenguaje.

Los equipos de trabajo son al máximo de dos personas y la entrega debe constar de un informe el cual debe contener, con el análisis del problema (se deben incluir premisas y requerimientos), el diseño de la solución (que incluya la descripción, diagramas y justificación de las estructuras de datos y algoritmos utilizados), detalles de la implementación (descripción de las estructuras de datos y funciones), casos de pruebas y los fuentes de programa (siendo importante el estilo y los comentarios). Así como un enlace a un repositorio con todos los archivos necesarios para compilar y ejecutar la aplicación.