Chapter 36 – Sparse Matrices

1. Sparse Matrices

Numeric matrices that mostly consist of elements with the value 0 present an opportunity for a programmer to both reduce the time spend by in running matrix multiplication and the space necessary to store the matrices.

A matrix element of 0 will always produce another 0 when multiplied. A matrix element with a value of 0 need not be included in the representation of the matrix at all. If the element is not present, its value is zero!

2. Storing Sparse Matrices

Each element of a matrix can be stored as a triple. That is, each element has three items of state information. These are the row, column and value of the matrix element. In order to save space when representing extremely large but sparse matrixes, a two-dimensional array consisting of three columns with as many rows as there are non-zero elements. For example, the matrix

$$\begin{vmatrix} 5 & 0 & 0 & 2 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 4 & 0 & 0 \end{vmatrix}$$

can be represented by the following array:

		Columns			
		0	1	2	
Rows	0	6	7	9	*indicates rows, cols., elements*
	1	0	0	5	*indicates value 5 at (0,0)*
	2	0	3	2	*indicates value 2 at (0,3)*
	3	0	5	-1	*indicates value -1 at (0,5)*
	4	2	1	1	*indicates value 1 at (2,1)*
	5	2	2	6	*indicates value 6 at (2,2)*
	6	3	3	-3	*indicates value -3 at (3,3)*
	7	4	1	8	*indicates value 8 at (4,1)*
	8	4	5	3	*indicates value 3 at (4,5)*
	9	5	3	4	*indicates value 4 at (5,3)*

Row 0 contains the information about the matrix. The 6 in row 0, column 0 indicates that the matrix has 6 rows. The 7 in row 0, column 1 indicates that the matrix has 7 columns. The 9 in row 0, column 2 indicates that there are 9 non-zero elements in the array.

3. Multiplying Sparse Matrices

The same matrix multiplication formula used on fully stored matrices is used to multiply sparse matrices, but it must be applied differently.

$$C_{ij} = \sum_{k=0}^{n-1} A_{ik} \bullet B_{kj}$$

The limits of *i* and *j*, i.e. the number of rows and columns in the new matrix, must be determined. The smaller of the number of columns in matrix *A* or *B* (A[0][1] verses B[0][1]) determines the number of columns in matrix *C*. The smaller of the number of rows in matrix *A* or *B* (A[0][0] verses B[0][0]) determines the number of rows in Matrix *C*.

To multiply matrices *A* and *B* together, the values of *i* and *j* must be generated.

```
for (int i=0; i<limitOfI; i++)
    for (int j=0; j<limitOfJ; j++)
```

The limit of *k* (the value of *n*) must be determined, which is the number of columns in matrix *A* or the number of rows in matrix *B*, whichever is smaller (A[0][1] verses B[0][0]). This limit can then be applied in the multiplication. For each possible *k*, sparse matrices must be checked to see if *A[i][k]* exists and *B[k][j]* exists. If any do, the results of the multiplication can be summed and placed in *C[i][j]*, otherwise 0 is placed in *C[i][j]* .

```
for (int i=0; i<limitOfI; i++)
    for (int j=0; j<limitOfJ; j++) {
        sum = 0;
        for (int k=0; k<n; k++)
            if A[i][k] exists and B[k][j] exists,
            multiply them and add the result to sum
        C[i][j]= sum;
    }
```

Programming Assignment 36.1

Devise and use an algorithm to translate a matrix into sparse matrix notation. Output the resulting matrix.

Programming Assignment 36.2

Create a program to multiply the sparse matrix in the example on page 1 (of this chapter) times the following sparse matrix. Output the resulting fully stored matrix.

		Columns 0	1	2
Rows	0	5	5	6
	1	0	1	3
	2	0	4	-2
	3	1	3	1
	4	2	2	-4
	5	2	3	2
	6	3	1	5
	6	4	4	4

Programming Assignment 36.3

Devise and use an algorithm to translate a sparse matrix into a normal matrix. Demonstrate that it works by outputting one of the example sparse matrix as a fully represented matrix in table form.