Assignment Number: 4

Problem statement:

Sparse Matrix: Write a menu driven C++ program with class for Sparse Matrix. Write functions to perform Sparse Matrix operations as listed below

1. Read sparse matrix
2. Display sparse matrix
3. Add two sparse matrices
4. Find transpose using Simple transpose algorithm
5. Find transpose using Fast transpose algorithm

Compare complexity of simple and fast transpose using counter.

Objectives:

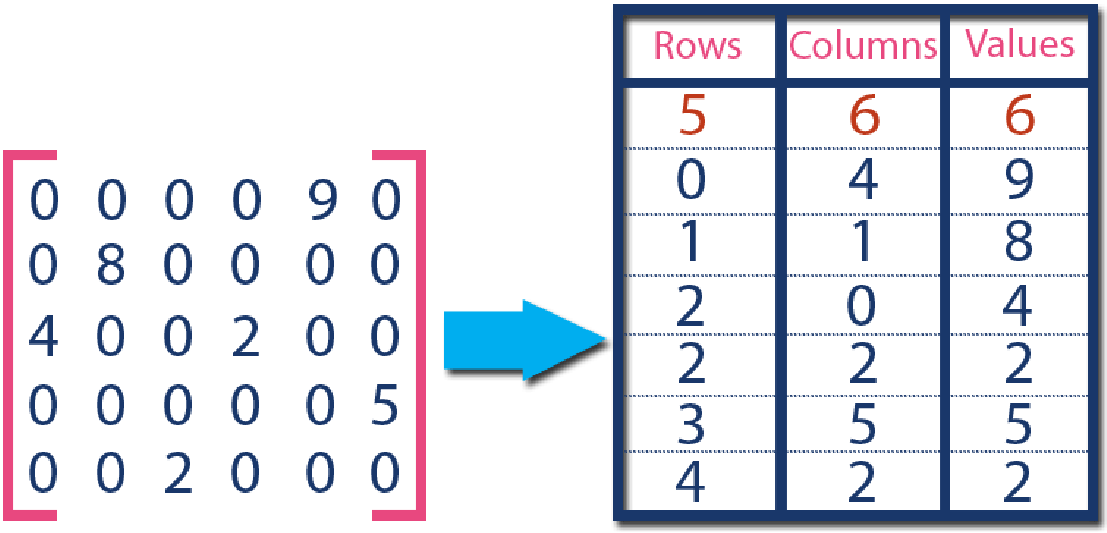
- Understand the use of array of structure
- To know the sparse matrix
- To know the difference between sparse matrix and normal matrix

Theory:

What is sparse matrix:

A sparse matrix or sparse array is a matrix in which most of the elements are zero. By contrast, if most of the elements are nonzero, then the matrix is considered dense.

Representation of sparse matrix:



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

Rows	Columns	Values
5	6	6
0	4	9
1	1	8
2	0	4
2	2	2
3	5	5
4	2	2

In above example matrix, there are only 6 non-zero elements (those are 9, 8, 4, 2, 5 & 2) and matrix size is 5 X 6. We represent this matrix as shown in the above image. Here the first row in the right side table is filled with values 5, 6 & 6 which indicate that it is a sparse matrix with 5 rows, 6 columns & 6 non-zero values. Second row is filled with 0, 4, & 9 which indicates the value in the matrix at 0th row, 4th column is 9. In the same way the remaining non-zero values also follows the similar pattern.

Algorithm:
For simple transpose of sparse matrix:

Input: Matrix sp1[ ][ ]
Output : Simple transpose matrix sp3[ ][ ]

```
Step 1: Start

Step 2: Read the non zero elements in matrix

Step 3: Read the no of rows and columns

Step 4: Read the nonzero elements with  row and column position

Step 5: store
            sp3[0][0]=sp1[0][1]
            sp3[0][1]=sp1[0][0]
            sp3[0][2]=sp1[0][2]
Step 5: for j=0 to sp1[0][1] and

        for i=1 to sp1[0][2]

         Check if   j=sp1[i][1]
         Copy
            sp3[k][0]=sp1[i][1];
            sp3[k][1]=sp1[i][0];
            sp3[k][2]=sp1[i][2];
         increment k
         increment i
         increment j
Step 6: Display resultant matrix

Step 7: Stop
```

Time complexity: Discuss the time complexity of following function with respect to best case and worst case

Simple transpose:

Fast transpose:

Practice problem:
Write a c++ code for sparse matrix subtraction and test your code for different test cases.