Name: Saurabh Mukherjee Roll no: 001910501006 Class: BCSE II Sem: First Session 2020-21 Assignment Set :2

Problem No. 1

Problem Statement :

Define an ADT for Sparse Matrix. Write C data structure representation and functions for the operations on the Sparse Matrix in a Header file. Write a menu-driven main program in a separate file for testing the different operations and include the above header file.

Solution Approach:

An array of structs is taken as the base structure where each field in the struct consists of the row number the column number and the value .We have the values of the matrix which are nonzero to be stored in the matrix in a linear fashion with the first matrix starting from the 0th index of the array and then the other matrix.Whenever we perform the addition we first check the compatibility of the addition by comparing the rows count and the columns count for both the given matrix and then add the corresponding matrix according to the matrix addition rules that is the corresponding column value is to be added with the same column and row value of the other matrix and hence we are getting a final matrix which is again to be stored in the same matrix and other operations which are to be performed too like multiplication are done in the same way.

Structured Pseudocode :

1.Define a structure as the sparse matrix with row,col and value as the date fields

2.Create another structure with an array of sparse matrix structure as its data field

3.Say Sparse\_mat is the name of the new adt we created

3.Initialise a=number of terms in first matric

4.Initialise b=number of terms in second matrix

5.Call Mutiply(Sparse\_mat ,a ,b)

6.For Add call add(Sparse\_mat,a,b)

7.for subtract call subtract(Sparse\_mat,a,b)

8.Transpose(Sparse\_mat,a,b)

Mutiply(Sparse\_mat \*m,a ,b)

1.for i from 0 to a

2. for j from a to b

3. if ((\*m)[i].col==(\*m)[j].row)

4. set result=(\*m)[i].value \* (\*m)[j].value

5. assign the result’s row as (\*m)[i].row and column as (\*m)[j].col

6. check the result matrix and find for row and col match both simultaneous

7. if match

8. add the result value with that value

9. else

10. make result as a new entry in the sparse\_mat

add(Sparse\_mat,a,b)

1.Initialise i=0,j=0

2.while i in range a and j from a to b

3. if the row and col both value matches

4. copy the value of row and col and add the corresponding values,increment both i,j

5. if the row of first matrix is lesser than that of second copy into result,increment i

6. if row of the second is lesser ,copy into result ,increment j

7. if rows are same, copy accordingly as the col , increment the corresponding loop counter

For subtraction the pseudocode is same only difference is the sign which would become negative instead of positive

Transpose(Sparse\_mat,a)

1.for i from 0 to a // a is the number of non-zero elements in the matrix

2. swap row and col value of the matrix

Results:

We get the result to be stored in the same Sparse\_mat array as the initial with the index values being modified and then later accessing the array to retrieve the date we just need to loop through it once .The result is basically stored in the same matrix which consists of both the sparse matrix along with the values .

Discussion:

We need to ensure that the size of the array taken inside the sparse matrix array is enough to hold the matrix non zero entries for both the matrix along with the result so that we do not run out of memory when storing the result values in the sparse matrix structure after the respective functions such as addition and multiplication and subtraction are called .We also need to pass the sparse matrix address inside the function so that the modifications made to the matrix upon storing the result in the same matrix is visible outside the function too when we access the matrix after the operation we get the values from the matrix which were a result of the matrix.

Separate files containing commented source code

The file has been attached.