Programming Assignment IV
Sparse Matrix

Due Date: 2024/10/23 (40) points

1 Description of the Assignment

Implement appropriate data structures to support efficient operations on sparse matrices. The operations should at least include addition (subtraction) and multiplication of matrices.

2 Input

Provide at least the following method for input.

1. The dimensions of the matrix $m \times n$.
2. List of the nonzero elements of each row.

For example, let the matrix be

$$M = \begin{pmatrix} 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 & 0 \end{pmatrix}$$

The input of the above matrix will be:

```
4 5
3 3 5 4 0
3 5 4 7 0
0
2 2 3 6 0
```

The first line of the input is the dimension of the matrix, m and n. In the above example, the dimension of the matrix is 4×5.

Each nonzero element is represented by c and v, where c is the column number of the nonzero element, and v is the value of the element. For each row of the matrix, a list of nonzero elements is given for that row, and the list is terminated by a 0. For example, for the last row (row 4),

```
2 2 3 6 0
```

shows that $M[4][2] = 2$, $M[4][3] = 6$. Note that row and column numbers must start from 1.

3 Output

Provide at least the following types of output for a matrix:

1. the regular format, i. e. m rows and n columns.
2. the list format, followed by the amount of the memory to store he matrix.

You may assume that all elements of the matrix are an integer. The amount of memory required to store a matrix includes the pointers, the column number and the value of the element. You may assume all these data can be stored in 1 memory unit.

4 Operations

In addition to the input and output of matrices, Implement at least the following operations on these matrices.

1. Addition of two matrices with the same dimension $m \times n$ in $o(mn)$ time.
2. Multiplication of two matrices of dimension $l \times m$ and $m \times n$ in $o(lmn)$ time.

Some other operations on matrices, such as transpose, may also be required, and should be implemented properly.

Assume that the number of nonzero elements in each input matrix is only linear in terms of m and n. Your program should use at most $O(e)$ spaces for all operations, where e is the number of nonzero elements in a matrix. That is, you cannot "expand" the matrix into $m \times n$ entries in memory. In addition, each row of the matrix should be stored in a separate list. Thus, for a matrix with m rows and n columns, the matrix should be represented by two integers m, n, and m lists. Matrices cannot be converted to regular 2-D arrays in all operations.

A new data type (or object) `SpaceMatrix` should be defined and implemented in your program, so that you can declare

```
SpaceMatrix M;
```

in you program. Then write a main program to demonstrate the usage of the new data type. For example, read two matrices A and B, print out the two matrices, and then print out $A + B$ and $A \times B$. If $A + B$ or $A \times B$ cannot be performed, print out error messages.

Notes

The format of the report of the assignment should be close to a technical research report, and include at least the following items:

1. Title and Author.
   This section should include assignment number, *your name*, *student number* and *email address* on the *first* part of the *first* page of your report.

2. Description of the Problem.
   A "formal" description of the problem in this assignment.
   In addition to the basic requirements specified in the assignment, highlight additional functions or features that you have implemented.
   Do not copy assignment instructions directly into this section.

3. Main results.
   This section should include at least the following items.

   (a) The design of your program.
   If the design or part of the design was obtained by reference to other sources, discussions, etc., appropriate citations should be given.

   (b) The data structures used in the program to improve the efficiency of the program.
   These data structures should be implemented by you and appear in the first part of your program.

   (c) List of your program with comments.

      i. If your program is very long, list only the main parts of the program here and the entire program in appendix.

      ii. Additional comments can be added manually to explain the design of the program.

   (d) Outputs of the compilation and the executions of your program.
   This section should use screen dumps whenever possible.

4. The performance evaluation of your program.

   (a) List the execution time of your program with various input sizes, such as $n = 100, 200, \cdots, 1000$.

   (b) The maximum input size that your program can run in a reasonable amount of time, such as 1 minute, 5 minutes, or 10 minutes.

5. Conclusions

   (a) Summarize what you did and the interesting things you learned from this assignment.

   (b) Describe the difficulties you encountered during program development and how you overcame them. (This is strong evidence that you did the work yourself.)

Additional notes:

1. Turn in your report on or before due day.

2. The output of the program execution should indicate the correctness of your program. In other words, a set of comprehensive (but not necessarily exhaustive) annotated test data for the problem should be provided to show that your program is indeed correct. This can be done by carefully selecting a set of test data.

3. Print or write the report on A4 paper and staple it to the upper left corner.