# Programming Assignment IV
# Sparse Matrix

Due Date: 2025/10/15 (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 $4 \times 5$.

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 the 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. Adding two matrices of the same dimension $m \times n$ takes $o(mn)$ time, e.g., $O(m + e)$ time, where $e$ is the number of non-zero elements.
2. Multiply two matrices of dimensions $l \times m$ and $m \times n$ in $o(lmn)$ time, e.g., $O(ln + e)$ time, where $e$ is the number of non-zero elements.

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 each assignment report should be cloase to a technical research report and must include at least the following sections:

1. **Title and Author**
   On the first part of the first page, clearly include:
   - Assignment number
   - Your name
   - Student number
   - Email address
2. **Description of the Problem**
   Provide a clear and formal description of the problem of this assignment.
   In addition to the basic requirements given in the assignment, highlight any extra functions or features you have implemented. Do not simply copy the assignment instructions into this section.
3. **Main Results**
   This section should include at least the following:
   (a) Program Design.
   Explain the overall design of your program. If any part of the design was obtained from references, discussions with other people, or other sources, proper citations must be provided.
   (b) Data Structures.
   Describe the data structures you implemented to improve efficiency. These should be your own implementations and appear in the first part of your program.
   (c) Program Listing with Comments.
      i. If the program is long, include only the main parts in the report body, and place the complete program in the appendix.
      ii. Add explanatory comments where appropriate to clarify your design.
   (d) Program Outputs.
   Include compilation results and execution outputs. Whenever possible, provide screen dumps.
4. **Performance Evaluation**

   (a) Report execution times of your program with various input sizes, such as $n = 100, 200, \ldots, 1000$.
   (b) Indicate the maximum input size your program can handle within a reasonable time limit (e.g., 1, 5, or 10 minutes).

5. **Conclusions**

   (a) Summarize what you accomplished and any interesting insights gained from this assignment.

   (b) Describe the challenges you encountered during development and how you overcame them. (This provides strong evidence that you completed the work independently.)

## Additional Notes

1. Submit your report on or before the due date.

2. The program output should clearly demonstrate correctness. That is, provide a set of comprehensive (but not necessarily exhaustive) annotated test cases to show that your program works correctly.

3. Print the report on A4 paper and staple it in the upper-left corner.