CSC3100 Assignment 2

Important Notes:

1. The assignment is an individual project, to be finished on one's own effort.
2. **Submission deadline is 6pm Mar. 10, 2023 (Friday). The deadline is tight. No late submission is accepted!**
3. Please also submit your final **two programs** and rename them as "StudentID_A2_AddSparseMatrix.java" and "StudentID_A2_MultiplySparseMatrix.java", respectively on the blackboard. For example, a student whose Student ID is "120000001" should submit **two programs** named as "120000001_A2_AddSparseMatrix.java" and "120000001_A2_MultiplySparseMatrix.java". **You don't need to consider the consistency of class name and file name** since we won't run the code you submitted on the blackboard. File misnaming or no submission on the blackboard will lead to **5 demerit points.**
4. OJ website: http://oj.cuhk.edu.cn/. If you are off campus, please use VPN to access the OJ.
5. Access code: csc3100as2
6. Each student is only permitted to submit code to OJ **up to TEN times for each problem**. Only the **last submission** will be used in evaluation of assignment marks.
7. Plagiarism is strictly forbidden, regardless of the role in the process. Notably, ten consecutive lines of identical codes are treated as plagiarism. Depending on the seriousness of the plagiarism, 30%-100% of marks will be deducted.

Marking Criterion:

1. The full score of the assignment is 100 marks.
2. **We have two problems in this assignment with 5 test cases each. Each test case has 10 marks.**
3. Zero mark is given if: there is no submission; a normal submission fails both 10 test cases.

Running Environment:

1. The submissions will be evaluated in the course's OJ system running Java SE version 17 and Linux platform.
2. The submission is only allowed to import four packages of (java.lang.*; java.util.*; java.math.*; java.io.*) included in Java SDK. No other packages are allowed.
3. All students will have an opportunity to test their programs in the OJ platform prior to the official submission.
4. In the test, each program is required to finish within **30 seconds of time**, with no more than **64MB memory**. **This is a strict requirement measured in the server environment!**

Submission Guidelines:

1. Inconsistency with or violation from the guideline leads to marks deduction.
2. It is the students' responsibility to read this assignment document and submission guidelines carefully and in detail. No argument will be accepted on issues that have been specified clearly in these documents.

Problem Description:

Write two programs to read two sparse matrices from the input, **add/multiply** the two matrices, and print out the result matrix. For example, the input and output have the following formats.

```
3, 2
1 1:7 2:9
2 :
3 2:-5
2, 3
1 2:3 3:5
2 1:-1 2:3 3:-5
```

Here is an illustration of the example.

- The first line of the file (“3, 2”) represents the size of the matrix is 3 rows and 2 columns. **Please note that there is a space after the comma (,).**
- The second line starts with 1, which means the 1st row of the matrix. “1:7” means the 1st column of the row is 7; “2:9” means the 2nd column of the row is 9.
- The third line starts with “2”, which means the 2nd row of the matrix. There are no other elements except a “:”, which means all the columns of the row are “0”.
- The fourth line starts with “3”, which means the 3rd row of the matrix. “2:-5” means the 2nd column of the row is -5.
- All matrix elements that are not listed explicitly are treated as 0. **In other word, only non-zero elements are listed in row and column orders.**
- All matrix elements are integer values.

Therefore, the file represents the following matrix:

$$\begin{bmatrix} 7 & 9 \\ 0 & 0 \\ 0 & -5 \end{bmatrix}$$

Similarly, the next three lines represent the following matrix:

$$\begin{bmatrix} 0 & 3 & 5 \\ -1 & 3 & -5 \end{bmatrix}$$

Functional Requirement

1. Write a program called **AddSparseMatrix.java** that reads two matrices (**of the same size**) from the input, add the two input matrices, and output the result matrix.

For example, the input is as follows (**two matrices with the same size**):

```
3, 2
1 1:7 2:9
2 :
3 2:-5
3, 2
1 1:7 2:9
2 :
3 2:-5
```

The corresponding output (the result matrix) is as follows:

```
3, 2
1 1:14 2:18
2 :
3 2:-10
```

2. Write a program called **MultiplySparseMatrix.java** that reads two matrices from the input, multiply the two input matrices, and output the result matrix. (Surely, the column number of the first matrix equals the row number of rows of the second matrix).

For example, the input is as follows (**two matrices with the same size**):

```
3, 2
1 1:7 2:9
2 :
3 2:-5
2, 3
1 2:3 3:5
2 1:-1 2:3 3:-5
```

The corresponding output (the result matrix) is as follows:

```
3, 3
1 1:-9 2:48 3:-10
2 :
3 1:5 2:-15 3:25
```

Note that, in the test:

1. The input matrices are **sparse**. Most matrix elements in **test cases** are zero. All non-zero elements are **integers**.
2. For **AddSparseMatrix.java**, the size of both input matrices will be no larger than 5000×5000.
3. For **MultiplySparseMatrix.java**, the size of first input matrices will be no larger than 500×10000 and the second will be no larger than 10000×500.