CS2311 Computer Programming, (18-19, Sem. A)

Assignment 2

Due: Nov 4, 2018, 23:59 p.m.

Late submission is not accepted. Plagiarism check will be performed.

Problem1: Sparse Matrix Storage

A sparse matrix is a matrix whose most elements are zero. It's not efficient to store the matrix as arrays with large memory footprint. Instead, we consider to store the matrix with the triple structure (i, j, v), where i denotes the index of row, j denotes the index of column, and v denotes the non-zero value in position (i, j). Specifically, the triples are stored in row-first, column-second order. An example is shown as follows:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 6 \\ 5 & 0 & 0 \end{bmatrix} \longrightarrow$$

(i, j, v)
(1,1,1)
(2,3,6)
(3,1,5)

Requirements:

(1) Given a matrix A $(n * m)$, print out its triple list. (One triple per line, in row-first, column-second order)

(2) Print out the triple list of the transpose of matrix A. (Format and order are same as question (1))

(3) Sort the triples by the **value** of the elements (in the ascending order) and print the sorted list.

(4) The input size n, m satisfy 0 < n, m < 10, and the element value is in the range of integer. You DO NOT need to consider any invalid input.

Expected Outputs: (The input data is in red. The output data is in black.)

Example-1

```
Please enter the number of rows and columns of the matrix:
3 3
Please enter the matrix:
1 0 2
0 5 0
4 0 0
The triple list of matrix is:
1 1 1
1 3 2
```

```
2 2 5
3 1 4
The triple list of transpose of matrix is:
1 1 1
1 3 4
2 2 5
3 1 2
The sorted triple list of matrix is:
1 1 1
1 3 2
3 1 4
2 2 5
```

Problem 2: Extremely Large Numeric Plus

Extremely large integer numbers cannot be represented as int, long, even int64_t types in C++. So how to store such extremely large integer numbers and perform addition and subtraction between two numbers?

Requirements:

(1) Define a class to store the extremely large integer as a character array.

(2) Achieve the functions of plus to calculate the **sum** of two big integers a and b. The input big integers a, b satisfy 0 <= a, b <= 10^100.

(3) P.S. All inputs are valid. You DO NOT need to check the data (e.g. negative numbers, invalid operation, prefix '+', etc.).

Expected Outputs: (The input data is in red. The output data is in black.)

Example-1
Please enter the equation: 1234567891011 + 1000000000000 The result is: 2234567891011