

1,

Data structure for a sparse matrix, I choose to use a nested map, to be specific, it is:

**map<int, map<int, double>> matrix**

the first int of the outer map stores the row number, while the second int of the inner map holds the column number, and the double could represent the values of the position at certain row and certain column (i, j).

how to find the corresponding value given row and column:

use the method "count" of map for the outer map to find whether the given row is existed or not, if not, return 0, if it is existed then use the method "count" of map for the inner map to find whether the given column is existed or not, if not, return 0, if it is existed then return the value at **matrix** [given row][given column], the time complexity is  $O(1)$

how to insert and delete the sparse matrix table entries given row and column:

for insert, **matrix** [given row][given column] = given value,

for delete, almost the same way as described above for finding, if existed then remove the given value.

the time complexity is  $O(1)$

2, implementation of readFromFile():

Every time read a line from the given file, for every line which contains three values: row, column and value, they could be stored as int, int and double, namely, insert into the matrix: setValue() method is called here

For data4\_4.txt file, which contains more than one million lines, it takes about 10 seconds to read the file

The size of the sparse matrix: it about 1 million rows and 1 million columns

How many non-zero elements does it have: 1001799

How to count it: declare a counter which increment one when read a more line

How many rows have no non-zero elements: 468864

How many columns have no non-zero elements: 468587

3, implementation of the matrix addition and subtraction, given matrix mat01 and mat02:

For addition: iterate the rows of mat01, and for every row which also contains a map, then iterate the columns of the row, do the same for mat02, namely, for the same row and column, if mat01 and mat02 both have value, add them, if only one have then set the other as 0 and add them

For subtraction, almost the same operation as addition, but use minus in place of plus.

4, implementation of the matrix multiplication:

Given matrix mat01 and mat02,

For the matrix mat03 of mat01 multiplying mat02, theoretically, it needs every row of mat01 times every column of mat02 to get mat03, while it takes too much of time and mat01 and mat02 are sparse matrices.

Here is my simplified solution of multiplication:

Firstly, get the transpose matrix of mat02, named transpose\_mat02,

Iterate every row of mat02 and every row which contains a map<int, double> for columns and values, get the numbers of the columns which means the other columns are 0, therefore, for mat02, only the corresponding rows could be taken into account for operations, by this way there are plenty of time could be saved. Then, use the every row of mat01 and every row of transpose\_mat02 for convolution operation to finish the multiplication.

5,

```
Test 1:
    Assignment: pass
    Resizing: pass
    Addition: pass
Test 2: pass
Test 3
    Addition: pass
    Multiplication: pass
Test 4 (reference time = 2.741)
    Summation: 0 secs (x 0), pass
    Addition: 2.77 secs (x 1.01058), pass
    Subtraction: 5.156 secs (x 1.88107), pass
    Multiplication: 7.27 secs (x 2.65232), pass
请按任意键继续. . .
```

My computer is HP: i5, 8G, 256 SSD.