# NSYSU-MATH Data Structure – Spring 2024

## Homework 2

## Design: Designing a SparseMatrixLL Class

### Data Preparation

For this assignment, you will find a zip file named `HW2.zip` containing template files and public test data. Your task is to implement a `SparseMatrixLL` class in either Python or C++. The directory structure and contents are as follows:

1. Python Implementation (`Py/` directory):
   - ✓ `SparseMatrixLL.py`: Implement your `SparseMatrixLL` class here.
   - ✓ `test.py`: Contains public test cases for your implementation.
2. C++ Implementation (`Cpp/` directory):
   - ✓ `SparseMatrixLL.cpp`: Implement your `SparseMatrixLL` class here.
   - ✓ `SparseMatrixLL.h`: The header file for your `SparseMatrixLL` class.
   - ✓ `main.cpp`: Contains public test cases for your implementation.

### Description

This assignment is divided into two main parts:

1. Class Implementation:
   - ✓ Implement a new class named `Polynomial` in the provided template file. For Python, use `Polynomial.py`. For C++, use `Polynomial.cpp` and `Polynomial.h`.
   - ✓ The specifications for the `Polynomial` class will be provided in the subsequent sections.
2. Discussion:
   - ✓ Discuss the difference between array and linked list and their pros and cons.
   - ✓ Discuss the difference between the `SparseMatrixLL` and `SparseMatrix` mentioned in our class and their pros and cons.

### ADT

The ADT is defined as follows:

| **`Polynomial` ADT** |
| --- |
| **Data:** A list (vector) that stores each row as a linked list and in the node of each linked list stores the column number and the actual value. Two integers that records the number of rows and columns of matrix |

**Operation:**

**Initialize:** Creates a new sparse matrix with given number of columns and rows.

**Getter:** Get a specific element from the matrix using row and column index: `m(r,c)` (Python) or `m.get(r,c)` (C++)

**Setter:** Set a value to specific location sing row and column index: `m(r,c)=v` (Python) or `m.set(r,c,v)` (C++)

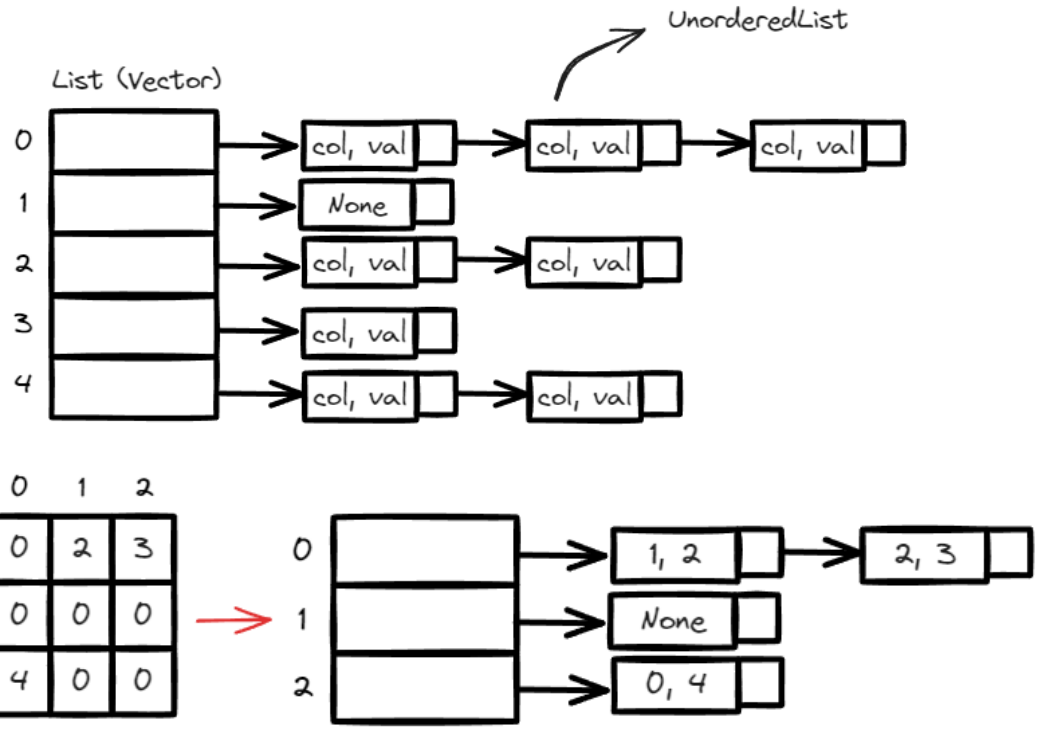
**Addition:** Add two sparse matrices and return the resulting matrix.

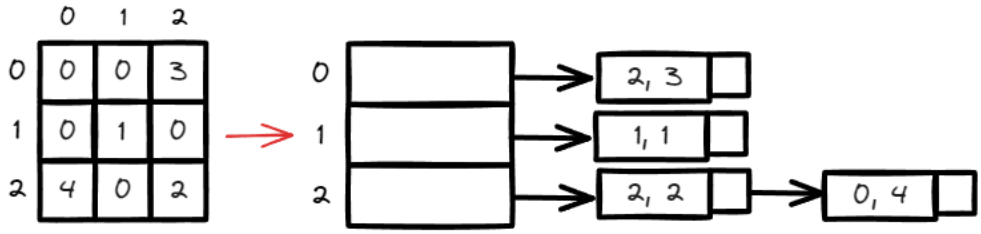
**Subtraction:** Subtract one matrix from the other and return the resulting matrix.

**Multiplication:** Multiply two sparse matrices and return the resulting matrix.

## Specifications

1. Class name: `SparseMatrixLL`
2. Attribute name: `_row_list, _nrow, _ncolumns` (They should be private)
3. Method: Constructor (nrows, ncolumns), getter, setter, +, −, ×. You should implement arithmetic operations using operator overloading. Note some custom methods for the class is already implemented. Do not modify these method.
4. **Use the `list` (in Python) or a `vector` (in C++) to store the rows. Each row of the matrix should be stored in a separate `UnorderedList`. Thus, for a matrix with $m$ rows and $n$ columns, the matrix should be represented by two integers `_nrow`, `_ncolumns` and $m$ `UnorderedList`.**
5. **The structure is described below, where the `data` of node is a `MatrixEntry` that contains `col` (the index of column) and `val` (the actual value):**






6. Please remove the item whose value is zero after operations.
7. The input values will be integers and you need to check the shape of matrix are compatible with each other before the operations.
8. 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.
9. You can only use standard Python or C++ library.

## Deliverables

1. Deadline: 2024/4/07 (Sun.), 11:59 PM. Hand in the following two items to the cyber universities. Please see our Facebook group for the late policy and rules.
2. Report:
   - ✓ Explain the design of your program and the data structures used. Discuss what you have learned from completing this homework.
   - ✓ Discuss the difference between array and linked list and their pros and cons.
   - ✓ Discuss the difference between the `SparseMatrixLL`, `SparseMatrix` and coordinate format mentioned in our class and their pros and cons.
3. Program Source Files:
   - ✓ Submit your source files in a zip file. **Ensure that you follow the provided template files**.
   - ✓ Source File Comments: Each file must begin with three lines of comments indicating the Author, Date, and Purpose of the program. Include appropriate comments throughout your code for clarity.

## Grading Policy

- Function Correctness: 60% (45% for public test cases and 15% for hidden test cases).
- Report and discussion: 40%.

## Reference

1. https://www.geeksforgeeks.org/sparse-matrix-representation/