Data Structure HW #2

Part I Paper Work

1 (Exercise 1 of Chapter 2.1) Write a C++ function to overload operator < for class Rectangle…

2 (Exercise 2 of Chapter 2.3) Write a C++ function to compare two ordered lists.

3 (Exercise 3 of Chapter 2.3) Modify function *Add …*

4 (Exercise 5 of Chapter 2.3) Write a C++ function that multiplies two polynomials…

5 (Exercise 3 of Chapter 2.4) Write a C++ function to input and output a sparse matrix. There should be implemented by overloading the >> and << operators, You should …

6 (Exercise 5 of Chapter 2.6) Write an algorithm that takes two strings x,y and return either -1, 0 or +1 if x<y, x=y, or x>y respectively.

7 (Exercise 7of Chapter 2.6) Compute the failure function for each of the following patterns.

   a. a a a a b

   b. a b a b a a

   c. a b a a b a a b b

Note: You can reuse one-side used papers but must in A4 size. Please hand in your assignments to the TAs (R721, Applied Science &Technology Building) by the deadline.

Part II Programming Exercise

1 A matrix is a rectangular array of numbers. When the number of rows and the number of columns of a matrix are the same, we called the matrix a square matrix. Implement the below Matrix ADT that is extended from ADT 2.4 in page 97 in C++.

class *Matrix*
{ **//** 三元組，<列，行，值>，的集合，其中列與行為非負整數，並且它的組合是
//唯一的；值也是個整數。

public:
Matrix(**int** *r*, **int** *c*,**int** *t*)**;**
// 建構子函式，建立一個有 *r* 列 *c* 行並且具有放 *t* 個非零項的容量
Matrix Transpose(**);**
//回傳將 ***this** 中每個三元組的行與列交換後的 *Matrix* A^T
Matrix Add(*Matrix b*)**;**
// 如果 ***this** 和 *b* 的維度一樣，那麼就把相對應的項給相加，

```
    // 亦即，具有相同列和行的值會被回傳；否則的話丟出例外。
    Matrix Multiply(Matrix b);
    // 如果*this 中的行數和 b 中的列數一樣多的話，那麼回傳的矩陣 d 就是 *this 和 b
    //（依據 d[i][j]=Σ(a[i][k]・b[k][j]，其中 d[i][j]是第 (i,j) 個元素）相乘的結果。k 的範
    // 圍 從 0 到*this 的行數減 1；如果不一樣多的話，那麼就丟出例外。
    int Determinant( );
    // 如果*this 是一個 square matrix，回傳det(A).
    int Adjoint( );
    // 如果*this 是一個 square matrix，回傳 adj(A).
    Matrix Inverse ( );
    // 如果*this 是一個 square matrix，回傳A^-1.
    Matrix Cofactor ( );
    // 如果*this 是一個 square matrix，回傳A_ij.
};
```

Note: In linear algebra, the **cofactor** A_{ij} is defined as the determinant of the square matrix of order $(n-1)$ obtained from *A* by removing the row number i and the column number j multiplied by $(-1)^{i+j}$. The **determinant,** denoted by $det(A)$ or $|A|$, is a value associated with a square matrix. The determinant of a square matrix A can be computed by a specific arithmetic expression as below.

$$\det(A) = \sum_{j=1}^{n} a_{ij}A_{ij}$$

Example 1.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 8 & 1 \\ 3 & 1 & 2 \end{bmatrix}, \det(A) = -56$$

Example 2.

$$B = \begin{bmatrix} 1 & 2 \\ 5 & 8 \end{bmatrix}, \det(B) = -2$$

The **adjoint** of a matrix A, denoted $adj(A)$, to be the transpose of the matrix whose ijth entry is A_{ij}.

Example 3.

$$adj\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} d & -c \\ -b & a \end{pmatrix}^T = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Example 4.

$$A = \begin{pmatrix} 1 & 3 & 2 \\ -1 & 0 & 2 \\ 3 & 1 & -1 \end{pmatrix}.$$

$$adj(A) = \begin{pmatrix} -2 & 5 & -1 \\ 5 & -7 & 8 \\ 6 & -4 & 3 \end{pmatrix}^T = \begin{pmatrix} -2 & 5 & 6 \\ 5 & -7 & -4 \\ -1 & 8 & 3 \end{pmatrix}.$$

The **inverse,** denoted by A^{-1}, is defined as

$$A^{-1} = \frac{adj(A)}{\det(A)}$$

Example 5.

$$\mathrm{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\mathrm{A}^{-1} = \frac{1}{\det \mathrm{A}} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Example 6.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} = \frac{1}{-2} \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix}$$

Note: You also have to write a C++ main program to test each class member functions. Please compress your code as well as the snapshot your execution results into a zip file and upload it to **iLearning**.