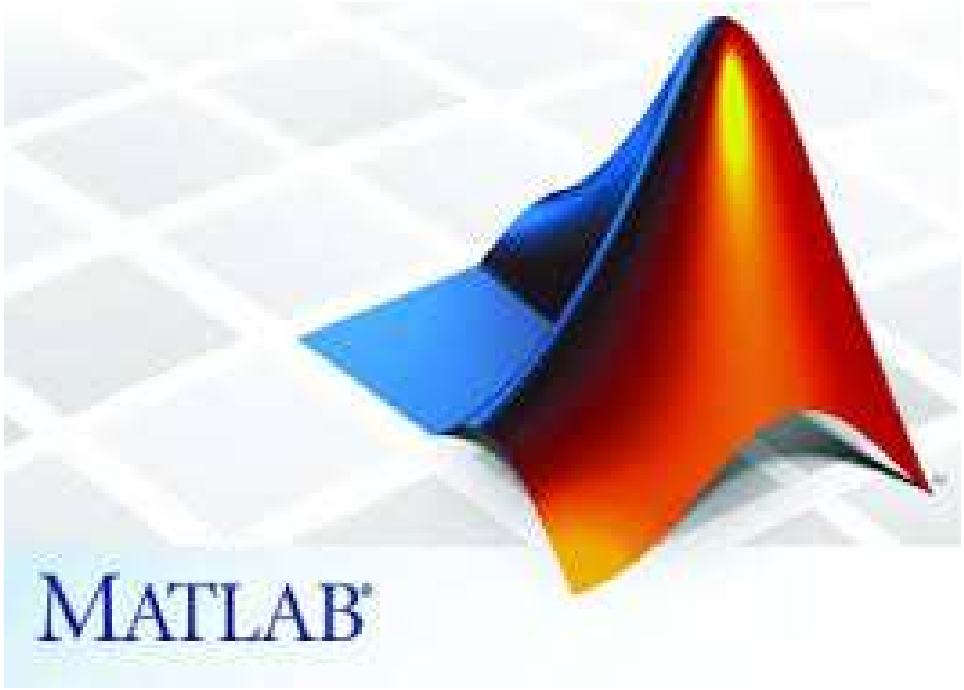
MATLAB을 선택한 이유



- 모바일 확장성
- Data structure, Computing Science 강력한 지원

Assignment (Data Structures : Array)

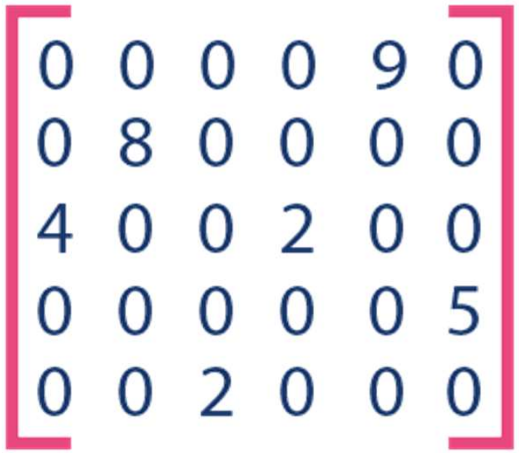
Write a C++ program to

1. Read two sparse square matrices, where two matrices are given as
   n : size of matrix – matrix A
   m : number of non-zero elements

2. Multiply them without converting to normal matrix

3. Convert the result matrix to normal matrix and write it as

행렬의 종류



Rows	Columns	Values
5	6	6
0	4	9
1	1	8
2	0	4
2	3	2
3	5	5
4	2	2

행렬 (Nomal) Matrix

30개

희소행렬 sparse Matrix

21개

과제 선택 이유 : 실제 환경의 사용

실제 환경에서 MATLAB에서 성능을 검증 하고 C++로 변환하는 작업에 이용

MATLAB, Python, Julia 스크립팅언어로 골격을 생성, 병목현상이 일어나는 부분만 C/FORTRAN으로 변환하여 포팅

MATLAB와 C++, Python을 상황에 따라 병용해서 사용

1. Sparse Matrix의 표현

```
>> A = sparse([0 2 0 1 0; 4 -1 -1 0 0; 0 0 0 3 -6; -2 0 0 0 2; 0 0 4 2 0]);
>> A

A =

   (2,1)        4
   (4,1)       -2
   (1,2)        2
   (2,2)       -1
   (2,3)       -1
   (5,3)        4
   (1,4)        1
   (3,4)        3
   (5,4)        2
   (3,5)       -6
   (4,5)        2
```

2. Sparse Matrix의 곱

```
>> x=A*A

x =

    (1,1)        6
    (2,1)       -4
    (3,1)       -6
    (5,1)       -4
    (1,2)       -2
    (2,2)        9
    (4,2)       -4
    (1,3)       -2
    (2,3)        1
    (3,3)      -24
    (4,3)        8
    (2,4)        1
    (3,4)      -12
    (4,4)        2
    (5,4)       12
    (1,5)        2
    (2,5)        6
    (3,5)        6
    (5,5)      -20
```

3. 희소행렬의 비희소행렬변환

```
>> B=full(x)

B =

     6    -2    -2     0     2
    -4     9     1     1     6
    -6     0   -24   -12     6
     0    -4     8     2     0
    -4     0     0    12   -20
```

결과

명령

```
>> A=sparse([0 2 0 1 0; 4 -1 -1 0 0; 0 0 0 3 -6; -2 0 0 0 2; 0 0 4 2 0]);

>> X=A*A;

>> B=full(X)

B =

     6    -2    -2     0     2
    -4     9     1     1     6
    -6     0   -24   -12     6
     0    -4     8     2     0
    -4     0     0    12   -20

>>
```

main.cpp | Sparse Matrix Multiplication.cbp | Sparse Matrix Multiplication.layout

```cpp
#include<iostream>
using namespace std;

#define MaxElements 100

class MatrixElement {
public:
    int row,col;
    int value;
};

class SparseMatrix {
public:
    int nRows,nCols,nElements;
    MatrixElement smArray[MaxElements];
    SparseMatrix(int r, int c): nRows(r), nCols(c), nElements(0) {
        nRows=r;
        nCols=c;
        nElements=0;
    }
    addElement(int r, int c, int v) {
        smArray[nElements].row=r;
        smArray[nElements].col=c;
        smArray[nElements].value=v;
        nElements++;
    }
    print() {
        cout<<"Sparse Matrix is " <<endl;
        for(int i=0;i<nElements;i++)
            cout<<smArray[i].row<<","<<smArray[i].col<<","<<smArray[i].value<<endl;
    }
};


int main () {
    int row = 4, col = 4;
    int a[row][col] = { {0, 0, 9, 0} , {5, 0, 8, 1} , {7, 0, 0, 2}, {0, 0, 0, 1} };
    SparseMatrix sm(row,col);
    for(int i = 0; i < row; i++)
        for (int j = 0; j < col; j++)
            if (a[i][j] != 0)  sm.addElement(i,j,a[i][j]);

    cout<<"The matrix is:"<<endl;
    for(int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) cout<<a[i][j]<<" ";
        cout<<endl;
    }

    sm.print();

    return 0;

}
```

```
The matrix is:
0 0 9 0
5 0 8 1
7 0 0 2
0 0 0 1
Sparse Matrix is
0,2,9
1,0,5
1,2,8
1,3,1
2,0,7
2,3,2
3,3,1
```