행렬의 곱셈

Level 2

행렬의 곱셈은, 곱하려는 두 행렬의 어떤 행과 열을 기준으로, 좌측의 행렬은 해당되는 행, 우측의 행렬은 해당되는 열을 순서대로 곱한 값을 더한 값이 들어갑니다. 행렬을 곱하기 위해선 좌측 행렬의 열의 개수와 우측 행렬의 행의 개수가 같아야 합니다. 곱할 수 있는 두 행렬 A,B가 주어질 때, 행렬을 곱한 값을 출력하는 productMatrix 함수를 완성해 보세요.

#include<iostream>

#include<vector>

using namespace std;

vector<vector<int> >productMatrix(vector<vector<int> >A, vector<vector<int> >B)

{

//m \* n n \* l = m \* l

int mSize = A.size();

int lSize = B[0].size();

int nSize = A[0].size(); // or B.size();

vector<vector<int> >answer(mSize, vector<int>(lSize));

//cout << A.size() << " " << A[0].size() << endl;

//cout << B.size() << " " << B[0].size() << endl;

//cout << answer.size() << " " << answer[0].size() << endl;

for(int i = 0; i < mSize; i++)

{

for(int j = 0; j < lSize; j++)

{

answer[i][j] = 0;

for(int k = 0; k < nSize; k++)

{

answer[i][j] += (A[i][k] \* B[k][j]);

}

}

}

return answer;

}

int main()

{

vector<vector<int> >A{{1,2},{2,3}};

vector<vector<int> >B{{2,3},{3,4}};

vector<vector<int> > testAnswer = productMatrix(A,B);

for(int i=0;i<testAnswer.size(); i++)

{

for(int j=0;j<testAnswer[i].size(); j++)

cout<<testAnswer[i][j]<<" ";

cout<<"\n";

}

}

/\*

과정 설명

//m \* n n \* l = m \* l

int mSize = A.size();

int lSize = B[0].size();

int nSize = A[0].size(); // or B.size();

for(int i = 0; i < mSize; i++)

{

for(int j = 0; j < lSize; j++)

{

answer[i][j] = 0;

for(int k = 0; k < nSize; k++)

answer[i][j] += (A[i][k] \* B[k][j]);

2 \* 3 3 \* 4 = 2 \* 4

m \* n n \* l = m \* l 행렬이 만들어진다는 것과

반복문이 돌아야할 크기만 잘 정해주면 문제없다.

\*/