MD NAHIDUL HOQUE NIRAB 2018100000023

Assignment#5 - Minimum Cost Path Problem Implementation using Dynamic Programming in any of your preferred Programming Language (C/C++/Java)

Code:

#include<bits/stdc++.h>

using namespace std;

int getMin (int num1, int num2)

{

if(num1>num2)

{

return num2;

}

else

{

return num1;

}

}

void equals\_1(int i, int j, int matrix [10][10])

{

cout << "(" << i << "," << j-1 << ")";

i=i+0;

j=j-1;

T: while (i>0)

{

if (matrix [i-1][j]<matrix [i][j-1])

{

cout << "(" << i-1 << "," << j << ")";

i=i-1;

j=j+0;

goto T;

}

else if (matrix [i-1][j]==matrix [i][j-1])

{

cout << "multiple path found: ";

cout << endl ;

cout << "1.";

equals\_1(i,j,matrix);

cout << endl;

cout << "2. ";

cout << "(" << i-1 << "," << j << ")";

i=i-1;

j=j+0;

goto T;

}

else

{

cout << "(" << i << "," << j-1 << ")";

i=i+0;

j=j-1;

goto T;

}

}

cout << "path end";

}

void possible\_path (int i, int j, int matrix [10][10])

{

i=i-1;

j=j-1;

cout << "(" << i << "," << j << ")";

T: if (i>0)

{

if (matrix [i-1][j] < matrix [i][j-1])

{

cout << "(" << i-1 << "," << j << ")";

i=i-1;

j=j+0;

goto T;

}

else if (matrix[i-1][j]==matrix[i][j-1])

{

cout << "multiple path found: ";

cout << endl ;

cout << "1.";

equals\_1(i,j,matrix);

cout << endl;

cout << "2. ";

cout << "(" << i-1 << "," << j << ")";

i=i-1;

j=j+0;

goto T;

}

else

{

cout << "(" << i << "," << j-1 << ")";

i=i+0;

j=j-1;

goto T;

}

}

cout << " (0,0) path end.";

cout << endl;

}

void minimum\_cost\_path(int c, int r, int matrix[10][10])

{

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

{

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

{

if (i==0 && j==0)

{

continue;

}

else if (i==0 && j!=0)

{

matrix[i][j] = matrix[i][j] + matrix[i][j-1];

}

else if (i!=0 && j==0)

{

matrix[i][j] = matrix[i][j] + matrix[i-1][j];

}

else

{

matrix[i][j] = getMin(matrix[i-1][j]+matrix[i][j], matrix[i][j-1]+matrix[i][j]);

}

}

}

cout<<"Length of minimum path sum: "<<matrix[r-1][c-1]<<endl;

possible\_path(r,c,matrix);

}

int main ()

{

int row, col;

cout << "Enter Row and Column of the matrix"<< endl;

cout << "Enter row = ";

cin >> row;

cout << "enter column= ";

cin >> col;

int matrix [10][10];

cout << " enter matrix : " << endl;

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

{

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

{

cout << "inputArray["<< i << "][" << j << "] = " ;

cin >> matrix[i][j];

}

cout << endl;

}

minimum\_cost\_path(col,row,matrix);

return 0;

}

Output:

