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
```

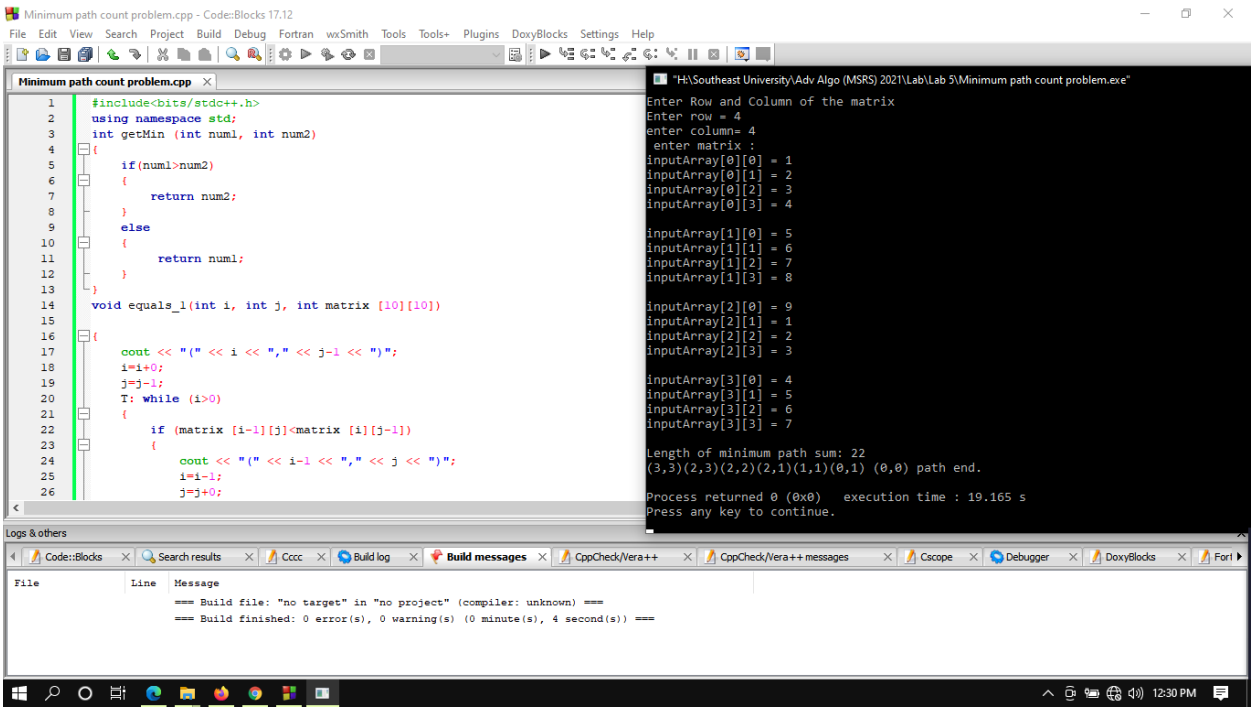
```
        {
            cout << "(" << i << "," << j-1 << ")";
            i=i+0;
            j=j-1;
            goto T;
        }
    }
    cout << "path end";

}
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:


Minimum path count problem.cpp - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Minimum path count problem.cpp
1 #include<bits/stdc++.h>
2 using namespace std;
3 int getMin (int num1, int num2)
4 {
5 if(num1>num2)
6 {
7 return num2;
8 }
9 else
10 {
11 return num1;
12 }
13 }
14 void equals_1(int i, int j, int matrix [10][10])
15
16 {
17 cout << "(" << i << "," << j-1 << ")";
18 i=i+0;
19 j=j-1;
20 T: while (i>0)
21 {
22 if (matrix [i-1][j]<matrix [i][j-1])
23 {
24 cout << "(" << i-1 << "," << j << ")";
25 i=i-1;
26 j=j+0;
"H:\Southeast University\Adv Algo (MSRS) 2021\Lab\Lab 5\Minimum path count problem.exe"
Enter Row and Column of the matrix
Enter row = 4
enter column= 4
enter matrix :
inputArray[0][0] = 1
inputArray[0][1] = 2
inputArray[0][2] = 3
inputArray[0][3] = 4
inputArray[1][0] = 5
inputArray[1][1] = 6
inputArray[1][2] = 7
inputArray[1][3] = 8
inputArray[2][0] = 9
inputArray[2][1] = 1
inputArray[2][2] = 2
inputArray[2][3] = 3
inputArray[3][0] = 4
inputArray[3][1] = 5
inputArray[3][2] = 6
inputArray[3][3] = 7
Length of minimum path sum: 22
(3,3)(2,3)(2,2)(2,1)(1,1)(0,1) (0,0) path end.
Process returned 0 (0x0) execution time : 19.165 s
Press any key to continue.
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
Cscope
Debugger
DoxyBlocks
File
Line
Message
=== Build file: "no target" in "no project" (compiler: unknown) ===
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 4 second(s)) ===
12:30 PM