Connected Cells in a Grid

by PRASHANTB1984

Consider a matrix with n rows and m columns, where each cell contains either a 0 or a 1 and any cell containing a 1 is called a *filled* cell. Two cells are said to be *connected* if they are adjacent to each other horizontally, vertically, or diagonally; in other words, cell $[i][j]$ is connected to cells $[i-1][j-1]$, $[i-1][j]$, $[i-1][j+1]$, $[i][j-1]$, $[i][j+1]$, $[i+1][j-1]$, $[i+1][j]$, and $[i+1][j+1]$, provided that the location exists in the matrix for that $[i][j]$.

If one or more filled cells are also connected, they form a *region*. Note that each cell in a region is connected to at least one other cell in the region but is not necessarily directly connected to all the other cells in the region.

Task
Given an $n \times m$ matrix, find and print the number of cells in the largest *region* in the matrix. Note that there may be more than one region in the matrix.

Input Format

The first line contains an integer, n, denoting the number of rows in the matrix.
The second line contains an integer, m, denoting the number of columns in the matrix.
Each line i of the n subsequent lines contains m space-separated integers describing the respective values filling each row in the matrix.

Constraints

- $0 < n, m < 10$

Output Format

Print the number of cells in the largest *region* in the given matrix.

Sample Input

```
4
4
1 1 0 0
0 1 1 0
0 0 1 0
1 0 0 0
```

Sample Output

```
5
```

Explanation

The diagram below depicts two regions of the matrix; for each region, the component cells forming the region are marked with an `X`:

```
X X 0 0     1 1 0 0
0 X X 0     0 1 1 0
0 0 X 0     0 0 1 0
1 0 0 0     X 0 0 0
```

The first region has five cells and the second region has one cell. Because we want to print the number of cells in the largest region of the matrix, we print 5.

Submissions: 8859
Max Score: 50
Difficulty: Medium

Rate This Challenge:
★★★★★ Thanks!

Need Help?
Depth First Search

More

Current Buffer (saved locally, editable)

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

class Solution {

     public class Celda
        {
            public int Fila;
            public int Columna;
            public int ColorCelda;


            public Celda(int fila, int columna)
            {
                this.Fila = fila;
                this.Columna = columna;
            }

            public override bool Equals(object obj)
            {
                Celda c = (Celda)obj;

                if (this.Fila == c.Fila && this.Columna == c.Columna)
                {
                    return true;
                }
                return false;
            }

            public override int GetHashCode()
            {
                return base.GetHashCode();
            }

        }


        static void mostrar(Celda[,] matriz, int filas, int columnas)
        {
            for (int i = 0; i < filas; i++)
            {
                for (int j = 0; j < columnas; j++)
                {
                    Console.Write(matriz[i, j].ColorCelda + " ");
                    //Console.Write(matriz[i, j].notacion + " ");
                }
                Console.WriteLine();
            }
        }

        public static List<Celda> FloodFill(Celda[,] matriz, int filas, int columnas, Celda nodo, int
viejo, int reemplazo)
        {

            Stack<Celda> pila = new Stack<Celda>();
            if (matriz[nodo.Fila, nodo.Columna].ColorCelda != viejo)
                return new List<Celda>();

            pila.Push(nodo);

            List<Celda> grupoSeleccionado = new List<Celda>();
            grupoSeleccionado.Add(nodo);


            while (pila.Count > 0)
            {
                Celda c = pila.Pop();

                matriz[c.Fila, c.Columna].ColorCelda = reemplazo;
                if (!grupoSeleccionado.Contains(matriz[c.Fila, c.Columna]))
                {
                    grupoSeleccionado.Add(matriz[c.Fila, c.Columna]);
                }

                if (c.Fila > 0)
                {
                    if (matriz[c.Fila - 1, c.Columna].ColorCelda == viejo)
                    {
                        pila.Push(new Celda(c.Fila - 1, c.Columna));
                    }
                }
                if (c.Fila < filas - 1)
                {
                    if (matriz[c.Fila + 1, c.Columna].ColorCelda == viejo)
                        pila.Push(new Celda(c.Fila + 1, c.Columna));
                }
                if (c.Columna > 0)
                {
                    if (matriz[c.Fila, c.Columna - 1].ColorCelda == viejo)
                        pila.Push(new Celda(c.Fila, c.Columna - 1));
                }
                if (c.Columna < columnas - 1)
                {
                    if (matriz[c.Fila, c.Columna + 1].ColorCelda == viejo)
                        pila.Push(new Celda(c.Fila, c.Columna + 1));
                }

                //----------diagonales------------

                if (c.Fila - 1 >= 0 && c.Columna - 1 >= 0)
                {
                    if (matriz[c.Fila - 1, c.Columna-1].ColorCelda == viejo)
                        pila.Push(new Celda(c.Fila - 1, c.Columna-1));
                }

                if (c.Fila - 1 >= 0 && c.Columna +1 < columnas)
                {
                    if (matriz[c.Fila - 1, c.Columna +1].ColorCelda == viejo)
                        pila.Push(new Celda(c.Fila - 1, c.Columna + 1));
                }

                if (c.Fila + 1 < filas && c.Columna + 1 < columnas)
                {
                    if (matriz[c.Fila + 1, c.Columna + 1].ColorCelda == viejo)
                        pila.Push(new Celda(c.Fila + 1, c.Columna + 1));
                }

                if (c.Fila + 1 < filas && c.Columna-1 >=0)
                {
                    if (matriz[c.Fila + 1, c.Columna - 1].ColorCelda == viejo)
                        pila.Push(new Celda(c.Fila + 1, c.Columna - 1));
                }

            }

            return grupoSeleccionado;

        }



        static void Main(string[] args)
        {

            int n = int.Parse(Console.ReadLine());
            int m = int.Parse(Console.ReadLine());

            int[,] tablero = new int[n, m];

            for (int i = 0; i < n; i++)
            {
                int[] linea = Array.ConvertAll(Console.ReadLine().Split(' '), e => int.Parse(e));

                for (int j = 0; j < linea.Length; j++)
                {
                    tablero[i, j] = linea[j];
                }
            }

            //for (int i = 0; i < n; i++)
            //{
            //    for (int j = 0; j < m; j++)
            //    {
            //        Console.Write(tablero[i, j] + " ");
            //    }
            //    Console.WriteLine();
            //}


            //int[,] tablero =
            //{
            //    {1, 1, 0 ,0},
            //    {0, 1, 1 ,0},
            //    {0, 0, 1 ,0},
            //    {1, 0, 0 ,0}
            //};

            int _filas = tablero.GetLength(0);
            int _columnas = tablero.GetLength(1);

            Celda[,] matriz = new Celda[_filas, _columnas];


            for (int i = 0; i < _filas; i++)
            {
                for (int j = 0; j < _columnas; j++)
                {
                    matriz[i, j] = new Celda(i, j);
                    matriz[i, j].ColorCelda = tablero[i, j];
                }
            }



            int max = 0;

            for (int i = 0; i < _filas; i++)
            {
                for (int j = 0; j < _columnas; j++)
                {
                    if (matriz[i, j].ColorCelda == 1)
                    {
                        List<Celda> sel = FloodFill(matriz, _filas, _columnas, new Celda(i, j), 1, 2);
                        max = Math.Max(sel.Count, max);

                    }

                }
            }

            Console.WriteLine(max);

           // Console.ReadLine();

        }

}
```

Line: 204 Col: 14

Upload Code as File

Test against custom input

Run Code

Submit Code

Congrats, you solved this challenge!

✓ Test Case #0 ✓ Test Case #1 ✓ Test Case #2

✓ Test Case #3 ✓ Test Case #4 ✓ Test Case #5

✓ Test Case #6

Next Challenge

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature