using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

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();

}

//static int ChessboardShapes(string[] squares)

//{

// int[,] m =

// {

// {0,1,0,1,0,1,0,1},

// {1,0,1,0,1,0,1,0},

// {0,1,0,1,0,1,0,1},

// {1,0,1,0,1,0,1,0},

// {0,1,0,1,0,1,0,1},

// {1,0,1,0,1,0,1,0},

// {0,1,0,1,0,1,0,1},

// {1,0,1,0,1,0,1,0},

// };

// for (int i = 0; i < 8; i++)

// {

// for (int j = 0; j < 8; j++)

// {

// matriz[i, j] = new Celda(i, j);

// matriz[i, j].ColorCelda = m[i, j];

// }

// }

// int f = 0, c = 0;

// for (int i = 8; i >= 1; i--)

// {

// for (int j = 'a'; j <= 'h'; j++)

// {

// matriz[f, c].notacion = (char)j + i.ToString();

// c++;

// }

// f++;

// c = 0;

// }

// for (int i = 0; i < 8; i++)

// {

// for (int j = 0; j < 8; j++)

// {

// if (squares.Contains(matriz[i, j].notacion))

// {

// matriz[i, j].ColorCelda = 1;

// }

// }

// }

// int max = 0;

// for (int i = 0; i < 8; i++)

// {

// for (int j = 0; j < 8; j++)

// {

// if (matriz[i, j].ColorCelda == 1)

// {

// List<Celda> sel = FloodFill(new Celda(i, j), 1, 2);

// max = Math.Max(sel.Count, max);

// }

// }

// }

// return max;

//}

}

}