COMP 2313 DATA STRUCTURES CODE ASSIGNMENT

|  |  |
| --- | --- |
| ***Assignment Name:*** *Island Count* | ***Student Name :****Mustafa S Topsakal* |
| ***Assignment Date:****11/30/2020* | ***Student id :****2704123* |

# Problem

# Given a 2D array binaryMatrix of 0s and 1s, implement a function getNumberOfIslands that returns the number of islands of 1s in binaryMatrix.

# An island is defined as a group of adjacent values that are all 1s. A cell in binaryMatrix is considered adjacent to another cell if they are next to each either on the same row or column. Note that two values of 1 are not part of the same island if they’re sharing only a mutual “corner” (i.e. they are diagonally neighbors).

# Explain and code the most efficient solution possible and analyze its time and space complexities.

# Code

class Main {

static final int ROW = 5, COL = 5;

boolean isSafe(int M[][], int row, int col,

boolean visited[][])

{

return (row >= 0) && (row < ROW) && (col >= 0) && (col < COL) && (M[row][col] == 1 && !visited[row][col]);

}

void DFS(int M[][], int row, int col, boolean visited[][])

{

int rowNbr[] = new int[] { -1, -1, -1, 0, 0, 1, 1, 1 };

int colNbr[] = new int[] { -1, 0, 1, -1, 1, -1, 0, 1 };

visited[row][col] = true;

for (int k = 0; k < 8; ++k)

if (isSafe(M, row + rowNbr[k], col + colNbr[k], visited))

DFS(M, row + rowNbr[k], col + colNbr[k], visited);

}

int countIslands(int M[][])

{

boolean visited[][] = new boolean[ROW][COL];

int count = 0;

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

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

if (M[i][j] == 1 && !visited[i][j])

{

DFS(M, i, j, visited);

++count;

}

return count;

}

public static void main(String[] args) throws java.lang.Exception

{

int M[][] = new int[][] { { 0, 0, 0, 1, 0 },

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

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

{ 0, 0, 0, 0, 0 },

{ 0, 0, 1, 1, 1 } };

Islands I = new Islands();

System.out.println("The number of islands is: " + I.countIslands(M));

}

}

# Outputs

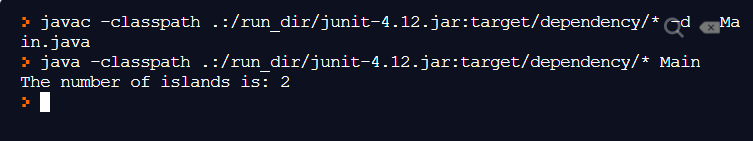


Figure Output Screen of the Odd Numbers Question

# REPL.IT LINK

[Repl.it - NiceSimilarBlogware](https://repl.it/@mustafasenih/NiceSimilarBlogware#Main.java)