Gebze Technical University
Computer Engineering

CSE 222 - 2019 Spring

HOMEWORK 3 REPORT

SILA BENGİSU YÜKSEKLİ
1801042877

Course Assistant: Erchan Aptoula

1 INTRODUCTION

1.1 Problem Definition

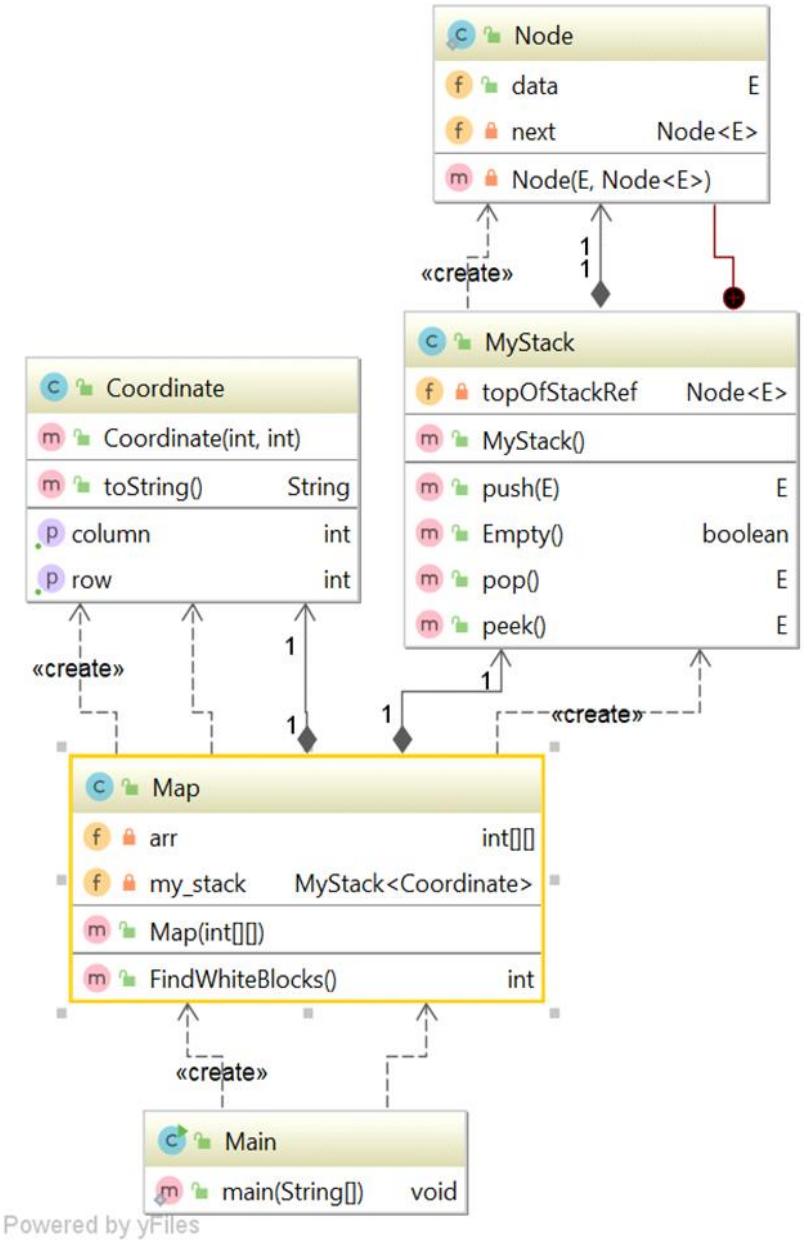
Some arbitary binary digital image is given. We don't know how many zeros and ones in this image and the places of them are uncertain. Some ones are friends of each other. Friend means if number of one which is located at some coordinate, has a neighboor which has a value of one and is located to the left , right , down or up of the our number, then they are friends of each other. So our aim is to make a group of ones that is friends to each other and calculate how many groups in this image.

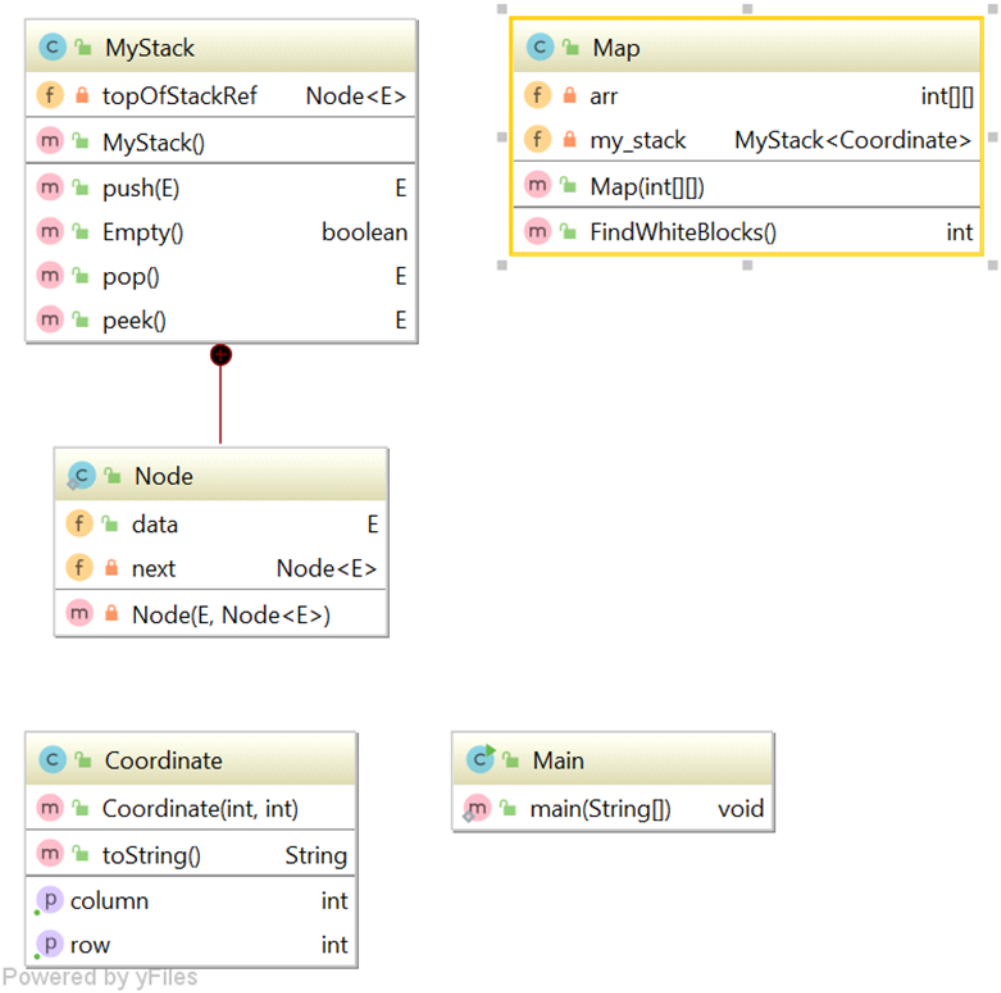
1.2 System Requirements

You can run this program everywhere which has java virtual machine. But if you want to use IntelliJ like me, maybe you need some requirements like minimum 1 GB RAM, 300 MB hard disk space, minimum 1 GB for caches, 1024x768 minimum screen resolutionSystem requirements.

2 METHOD

2.1 Class Diagrams





2.2 Use Case Diagrams

The user can open the file in intellij and run from the button which is located at right-up edge, but if user want to run in some other place, if there is java virtual machine, he/she can do.

2.3 Problem Solution Approach

After I had read the problem, I made a little research about depth first search, watched some videos and I realize our problem is very similar to the maze problems that I watched.

So first I implemented my own stack, because it is very useful for this problem. We can think our problem as a maze, all ones indicates a road that we can walk and all zeros, lets say indicates a wall. I thought respectively like this:

I am going to visit all numbers, but when I visit the first number of one you see on the map, I need to find its friends, if it has friends. Every time I visit the friends, I push them in the

stack as a coordinate and if there is no place to go in the end I need to go back and look is there any missing friend. While I was going back, I remove them from the stack. Finally after all the operations, stack will be empty. This process continues until stack is empty and when the stack is empty I will find a group of ones which called as white block.

I implemented stack as Linked List data structure, because I thought it is the easiest way. If I use vector or arrays, it would be a bad choice because all vector methods are accessible and I had to use add and remove operations all the time for this problem, so it would be very diffucult with arrays.

As I explained in the first queation, the problem was to find the number of the white blocks. Every ones which is in the white block, has a x and y coordinate , I mean x equals number of the row and y equals number of the column in our case. Because of using two dimensional arrays, my stack had to keep two components. That's why I made a class which is called Coordinate and my stack kept x,y coordinates as a Coordinate object.

The time complexity of my algorithm is O(n), n represents the row x column .

3 RESULT

3.1 Test Cases

I tested my program in main repeatedly by using some arbitary two dimesional arrays. First I calculated the result by myself, then I compare with the result that was found by computer. Then I wrote in file some other arrays and tested like that too. After like fifteen tests, I concluded my program works as expected.

3.2 Running Results

TEST CASE 1:

MyStack [C:\Users\sila_\MyStack] - ...\src\com\company\Main.java [MyStack] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

MyStack 〉 src 〉 com 〉 company 〉 Main

Main.java × | MyStack.java × | Coordinate.java × | Map.java ×

```java
package com.company;
import java.io.*;
import java.util.*;

public class Main {

    public static void main(String[] args) {


        int R_counter=0;
        int C_counter=0;
        int n=0;
        int i;
        int map[][]=null;

        try {

            String row_str;
            String[] col_str=null;
            File f = new File(args[0]);
            Scanner scan = new Scanner(f);


            while (scan.hasNextLine()) {
                row_str = scan.nextLine();
                col_str = row_str.split( regex: " ");
                ++R_counter;
            }


```

MyStack [C:\Users\sila_\MyStack] - ...\src\com\company\Main.java [MyStack] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

MyStack 〉 src 〉 com 〉 company 〉 Main

Main.java × | MyStack.java × | Coordinate.java × | Map.java ×

```java
                ++R_counter;
            }


            C_counter=col_str.length;
            map = new int [R_counter][C_counter];

            scan= new Scanner(f);

            while (scan.hasNextLine()) {
                row_str = scan.nextLine();
                col_str = row_str.split( regex: " ");
                for(i=0; i<C_counter;++i)
                    map[n][i]=Integer.parseInt(col_str[i]);
                ++n;
            }
        }

        catch (FileNotFoundException exception) {
            exception.printStackTrace();
        }


        Map m= new Map(map);
        int check= m.FindWhiteBlocks();
        System.out.print("Number of the white blocks are:");
        System.out.println(check);
    }
}
```

Main

input - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

```
0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0
0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
```

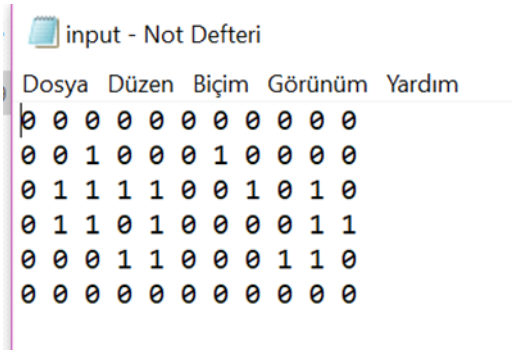
input.txt

Run: Main ×

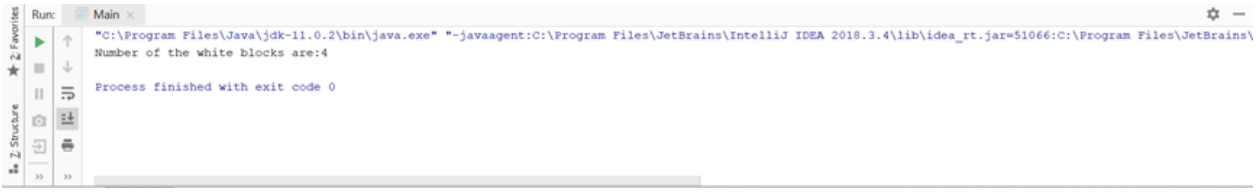
```
"C:\Program Files\Java\jdk-11.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2018.3.4\lib\idea_rt.jar=51032:C:\Program Files\JetBrains\
Number of the white blocks are:9

Process finished with exit code 0
```

Result

TEST CASE 2:

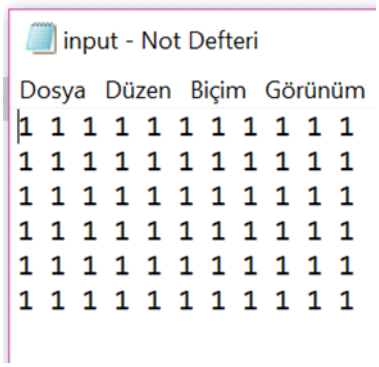


input.txt



result

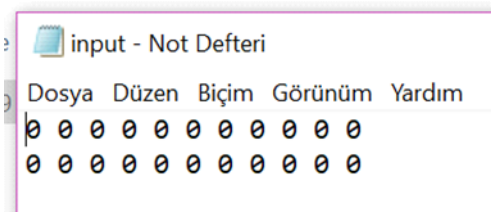
TEST CASE 3:



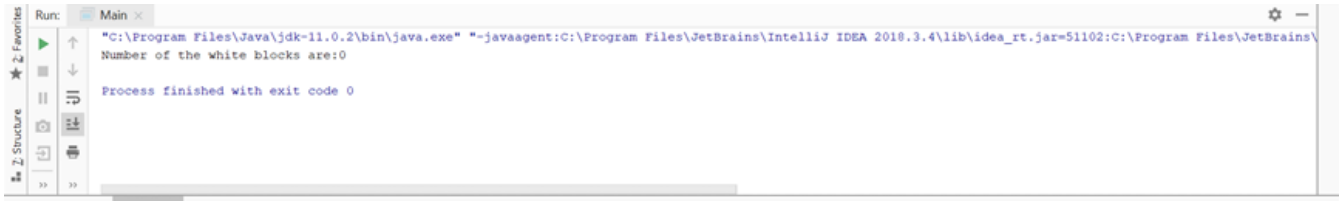
input.txt

result

TEST CASE 4:



input.txt



result