

CIS 2168: Assignment #6

Due on Monday, October 13, 2014

11:59pm

Anwar Mamat

Problem 1

List Folders (100%)

A binary digital image a rectangular grid of black and white pixels. The image shown in Figure 1 consists of 17*9 pixels.

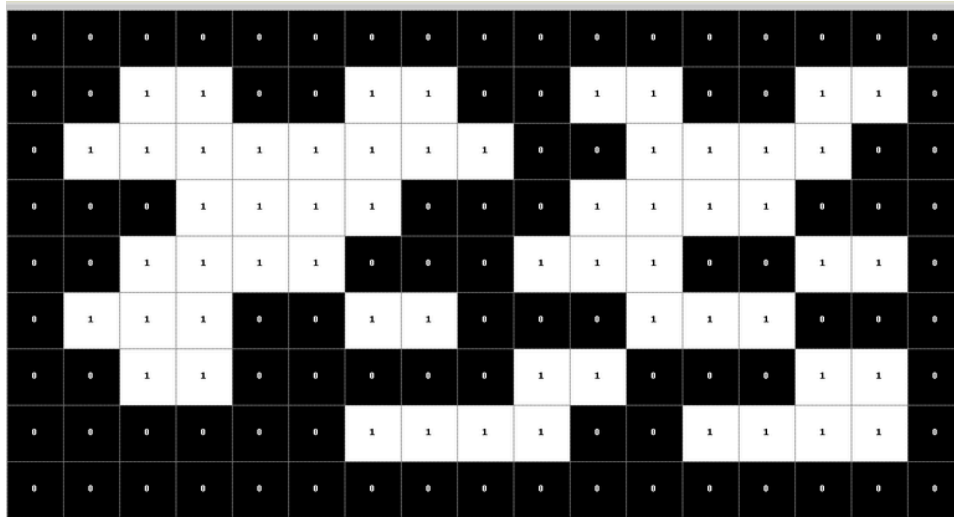


Figure 1: Image with 17 * 9 black and white pixels

In this assignment, you write a program to count the connected blocks in a binary digital image. For example: There are three connected blocks in Figure 2a, six connected blocks in Figure 2b, 14 connected blocks in Figure 2c, and one connected block in Figure 2d. If we count white blocks, there are two connected blocks in Figure 1.

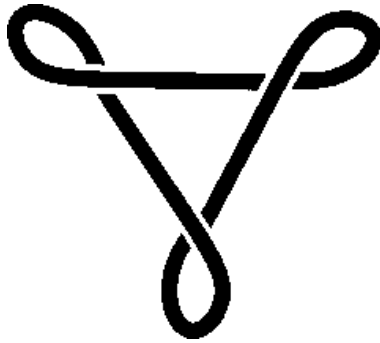
Create a NetBeans java project, add the jar file “stdlib-package.jar”. The “Picture” class in this package can read and show digital image files. You can also get the height, width of the image, and the color of each pixel on the image. The following code opens a image file, displays it, returns the height and width of the image, and the color of a pixel at grid position x,y.

```
Picture pic = new Picture(fileName);
pic.show();
height = pic.height();
width = pic.width();
5 Color t = pic.get(x, y);
```

In order to find a connected block, you find a black pixel, then you recursively visit all neighbors of the pixel to check if any neighbor is a black pixel too. Then you move to the neighbor, and repeat until no more “black pixel” neighbors. As shown in Figure 3, there are 8 neighbors a pixel.

You can use this skeleton code. Function “CountConnectedBlocks” receives a file name and returns the number of connected black-colored blocks.

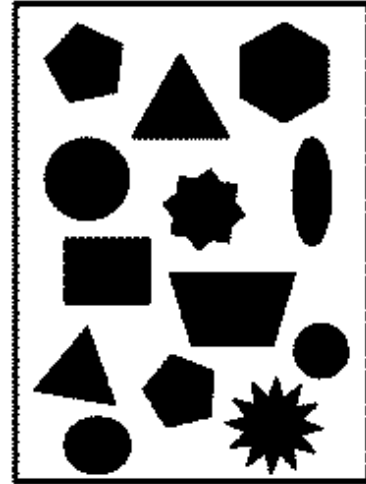
```
import edu.princeton.cs.introcs.Picture;
import java.awt.Color;
public class ImageBlocks {
    static Color BLACK = new Color(0,0,0);
    static Color WHITE = new Color(255,255,255);
    private int height;
    private int width;
    int [][] visited;
```



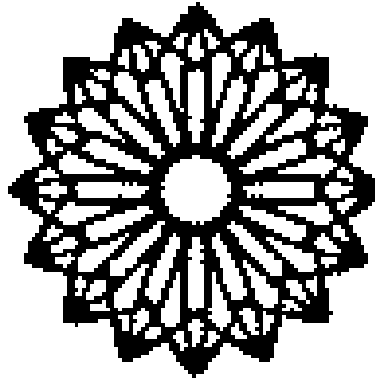
(a) image with 3 connected blocks



(b) image with 6 connected blocks



(c) image with 14 connected blocks



(d) image with 1 connected blocks

Figure 2: Binary digital images

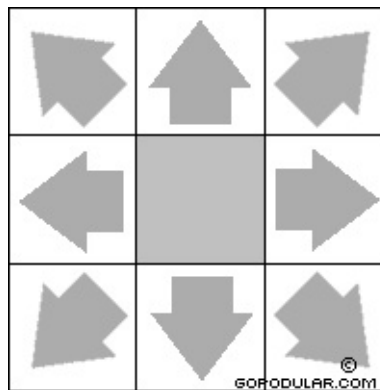


Figure 3: Neighbors of a pixel

10

```

Picture pic;
/**
 *   returns true if the pixel at (x,y) is black. false otherwise
 */

```

```
15 private boolean isBlack(int x,int y){
    return pic.get(x,y).equals(BLACK);
}
/**
 * returns true if the pixel at (x,y) is white. false otherwise.
 */
20 private boolean isWhite(int x,int y){
    return pic.get(x,y).equals(WHITE);
}
public int CountConnectedBlocks(String fileName){
    int count = 0;
25 pic = new Picture(fileName);
    pic.show();
    height = pic.height();
    width = pic.width();

30 // TO DO

    return count;
}

35 public static void main(String[] args) {
    String fileName = "image.png";
    ImageBlocks block = new ImageBlocks();
    try{
        int c = block.CountConnectedBlocks(fileName);
40 System.out.println("Number of connected blocks="+c);
    }catch(Exception ex){
        System.out.println(ex.getMessage());
    }
}
45 }
```

Testing

I included images to test. The file names and number of blocks are shown in Table 1. If you test the program with your other images, make sure the image is not too big. You may get “StackOverflow” error for images if the connected block is large too.

Implementation Requirements

You are allowed to use **java provided collections** if you need to.

What will you learn from this assignment

Recursion, Stacks

Table 1: Image files for test

File name	Number of black blocks
1.0.png	1
1.1.png	1
1.2.png	1
1.3.png	1
2.png	2
3.1.png	3
3.2.png	3
3.4.png	3
4.png	4
6.png	6
7.png	7
14.png	14

Grading

Homework is 100 points. 80 will reflect functionality and correctness. 20 points on your program will reflect your programming style, documentation. **If you code does not compile, you will not receive any credit. If you submit wrong file or empty project you will not receive any credit. Late assignments will receive zero.**

Commenting and Documenting Code

Code must be properly commented. The main idea is that the grader should be able to understand your code easily, not have to tear his or her hair out wondering what some statement is doing. The first time you have to deal with poorly commented code (if you haven't already), you will understand how annoying it is. In particular, the top of each code file should contain your name, the course and assignment numbers, and a brief summary of what's in the file. Line-by-line comments should be included as necessary to make the code easy to read. A clear coding style, together with informative variable and function names, will reduce the number of comments required. Obscure code or cryptic function names will cause loss of points (for bad style) and also require more extensive comments.

What to submit

A single zip file called Assignment6_firstname_lastname.zip, where firstname is your first name, and lastname is your last name. In this zip file, put:

1. Java source
2. A README file with:
 - Instructions to compile and run of your code (include a description of command line options).
 - If your solution is not perfect, explain what parts you did and what part you did not do.
 - List of files submitted
 - All your data and results (in plain text files).
 - Anything else you want TA know

3. Submit this zip file to Blackboard