**CS381-37: Project 2.1 (JAVA)**

**Yida Tao**

**Due date: Feb. 13, 2018**

Algorithm Steps:

step 0: - open the input file and output file

- read the image header, the four numbers

- dynamically allocate mirrorFramedAry

- dynamically alloicate tempAry

step 1: read the input file and load onto mirrowframeAry begin at [1,1]

step 2: mirrowFramed (mirrorFramedAry)

step 3: process the MirrorframedAry, from left to right and top to bottom

using i, and j, begin at (1, 1)

- neighborAry <- load MirrorframedAry[i,j]'s 3 X 3 neighborhoods

- tempAry[i,j] <-- compute the averaging of neighborAry

- keep tracking the newMin and newMax of tempAry

step 4: repeat step 3 until all pixels are processed

step 5: output the image header (numRows, numCols, newMin, newMax) to AVG3X3Out.txt.

step 6: output to AVG3X3Out.txt from tempAry, begin at [1,1], without the pixels on the boarder.

step 7: close input file and AVG3X3Out.txt

**Source Code**

**import** java.io.\*;

**public** **class** Project2 {

**public** **static** **void** main(String[] args) {

//set output to output file

FileOutputStream fos = **null**;

**try** {

fos = **new** FileOutputStream(args[1]);

System.*setOut*(**new** PrintStream(fos));

} **catch** (FileNotFoundException e) {

e.printStackTrace();

}

//step 0,1

AVG3X3 a = **new** AVG3X3(args[0]);

//step 2

a.mirrorFramed();

//step 3,4

a.Averaging();

//step 5,6

a.printTempA();

//step 7: close input file and output file

**try** {

fos.close();

System.*setOut*(**new** PrintStream(**new** FileOutputStream(FileDescriptor.***out***)));

} **catch** (IOException e) {

e.printStackTrace();

}

System.***out***.println("Done");

}

}

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.util.Scanner;

**public** **class** AVG3X3 {

**int** numRows = 0;

**int** numCols = 0;

**int** minVal = 0;

**int** maxVal = 0;

**int** newMin = 0;

**int** newMax = 0;

**int** mirrorFramedAry[][] = **null**;

**int** tempAry[][] = **null**;

**int** neighborAry[] = **null**;

//constructor

**public** AVG3X3(String input){

Scanner sc = **null**;

neighborAry = **new** **int**[9];

**try** {

sc = **new** Scanner(**new** File(input));

} **catch** (FileNotFoundException e) {

e.printStackTrace();

System.***out***.println("Cant find the file" + input);

}

**try** {

numRows = sc.nextInt();

numCols = sc.nextInt();

minVal = sc.nextInt();

maxVal = sc.nextInt();

mirrorFramedAry = **new** **int**[numRows + 2][numCols + 2];

tempAry = **new** **int**[numRows + 2][numCols + 2];

**int** counter = 0;

**int** r = 0;

**int** c = 0;

**while**(sc.hasNextInt()){

r = counter/numCols + 1;

c = counter%numCols + 1;

mirrorFramedAry[r][c] = sc.nextInt();

counter++;

}

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

sc.close();

}

}

**public** **void** mirrorFramed(){

**for**(**int** i = 1; i < numCols + 1; i++){

mirrorFramedAry[0][i] = mirrorFramedAry[1][i];

mirrorFramedAry[numRows + 1][i] = mirrorFramedAry[numRows][i];

}

**for**(**int** j = 0; j < numRows + 2; j++){

mirrorFramedAry[j][0] = mirrorFramedAry[j][1];

mirrorFramedAry[j][numCols + 1] = mirrorFramedAry[j][numCols];

}

}

**private** **void** loadNeighbors(**int** r, **int** c){

**int** index = 0;

**for**(**int** i = -1;i < 2; i++){

**for**(**int** j = -1; j < 2; j++){

neighborAry[index] = mirrorFramedAry[r+i][c+j];

index++;

}

}

**int** avg = AvgNeighbor();

**if**(avg < newMin){

newMin = avg;

}

**else** **if**(avg > newMax){

newMax = avg;

}

tempAry[r][c] = avg;

}

**private** **int** AvgNeighbor(){

**int** sum = 0;

**for**(**int** i = 0; i < 9; i++){

sum += neighborAry[i];

}

**return** sum/9;

}

**public** **void** Averaging(){

**for**(**int** i = 1; i < numRows + 1; i++){

**for**(**int** j = 1; j < numCols + 1; j++){

loadNeighbors(i,j);

}

}

}

**public** **void** printTempA(){

System.***out***.println(numRows + " " + numCols + " " + newMin + " " + newMax);

**for**(**int** i = 1; i < numRows + 1; i++){

**for**(**int** j = 1; j < numCols + 1; j++){

System.***out***.print(tempAry[i][j] + " ");

}

System.***out***.println();

}

}

}

**Output**

AVG3X3Out1\_PP:

31 40 0 9

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 2 2 2 2 2 2 2 1 1 1 2 2 2 1 2 2 2 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1

1 1 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 2 1 2 1 2 1 2 2 2 2 2 1 2 1 2 1 2 1 2 1 2 1 2 1 1 1 1 1 1 2 2 2 1

1 1 2 1 1 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 1 3 3 3 1

1 2 1 1 2 2 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 2 4 3 3 1

1 2 2 1 2 2 3 4 5 6 7 6 7 6 7 6 6 6 7 7 7 6 6 6 7 6 6 6 6 6 6 4 3 2 3 2 2 1 1

1 2 2 1 2 2 3 5 7 8 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 7 7 6 4 3 2 2 1 1 1 1

1 1 2 1 1 1 2 4 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 7 6 4 2 1 1 2 2 1

1 2 2 1 1 1 2 5 7 9 8 8 8 9 9 9 9 9 9 9 9 9 9 8 8 8 9 9 8 8 6 4 2 1 1 2 2 1

1 2 3 2 1 1 1 2 4 7 9 9 9 9 9 9 9 9 9 9 9 9 9 8 7 6 7 8 9 8 8 5 3 1 1 1 2 1 1

1 3 3 3 1 1 1 2 4 7 8 9 9 8 8 7 8 7 8 8 9 9 9 8 6 4 5 7 9 8 8 5 3 1 1 1 2 1 1

1 2 3 2 1 1 1 2 5 7 8 9 8 6 6 6 7 5 6 7 9 9 9 8 7 5 6 7 9 8 8 5 3 1 1 1 2 1 1

1 1 1 2 2 3 5 7 8 9 8 6 6 6 6 3 4 6 9 9 9 9 8 7 7 8 9 8 8 5 3 1 1 1 1 1 1

1 2 2 2 3 4 6 8 9 9 8 7 7 8 7 5 5 7 9 9 9 9 8 8 8 9 9 8 8 5 3 1 1 1 1 1

1 2 2 3 4 5 6 8 9 9 9 9 9 9 8 7 7 8 9 9 8 8 7 8 8 9 9 8 8 5 3 1 1 1 1 1

1 1 2 2 3 4 5 6 8 8 8 8 9 9 9 9 9 9 9 9 9 7 6 5 7 8 9 9 8 7 4 2 1 2 2 2 1 1

1 1 1 2 2 3 4 6 7 8 8 8 8 8 8 8 8 8 8 8 8 6 5 5 7 8 8 8 8 7 4 2 1 2 2 3 2 1

1 1 2 2 2 3 4 6 7 8 8 8 8 8 8 8 8 8 8 8 8 7 6 5 6 7 7 8 8 7 4 2 1 2 2 3 2 2

1 1 1 1 2 3 4 5 6 7 7 7 7 6 7 7 7 7 7 7 7 6 7 6 6 6 6 7 6 6 5 3 1 1 1 2 2 1

1 1 1 1 2 3 3 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 4 4 4 5 4 4 3 2 1 2 2 2 1 1

1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 1 1

1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 2 2 1 1

1 1 1 1 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 2 2 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 2 2 2 1 1

1 1 2 2 1 1 1 2 2 2 1 1 1 2 2 2 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 2 1 2 1 1

1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1

1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1

AVG3X3Out1\_thr\_7\_PP:

31 40 0 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1

AVG3X3Out1\_thr\_8\_PP:

31 40 0 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1

AVG3X3Out2\_PP:

31 40 0 8

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3 1 1 1

1 1 2 1 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 3 4 3 1 1 1 1 1 1 1 1 1 3 4 5 3 1 1 1 2

1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 3 5 6 6 3 2 1 1 1 1 1 2 2 1 3 4 5 3 1 1 1 2

1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 3 6 7 8 7 5 3 2 1 1 1 1 2 2 1 2 3 3 2 1 1 1 2

1 1 1 1 1 1 1 1 1 1 1 1 1 2 3 6 7 8 7 6 5 5 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 3 5 7 7 8 6 4 4 5 5 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 2 3 5 7 8 7 7 6 4 4 6 7 5 3 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 2 3 5 7 8 7 7 7 7 6 6 7 8 7 6 3 1 1 1 1 1 1 1 1 1 1 1

1 1 2 1 1 1 1 1 1 1 1 2 3 5 7 8 7 7 7 8 8 8 8 8 8 8 7 6 3 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 3 5 7 8 7 7 7 8 8 8 7 6 6 7 8 8 7 5 3 1 1 1 1 1 1

1 1 1 1 1 1 1 1 2 5 7 7 7 7 7 8 8 8 8 6 4 4 6 8 8 7 7 5 3 1 1 1 1 1

1 1 1 1 1 1 1 3 5 7 8 7 8 7 7 7 8 8 8 6 4 4 6 8 8 7 7 7 5 3 2 1 1

1 1 1 1 1 1 3 5 7 7 8 8 8 8 8 7 7 7 6 5 5 6 7 8 8 7 7 7 7 5 4 2 1 1

1 1 1 1 1 1 2 3 5 7 8 7 8 8 8 7 7 7 8 6 5 5 7 8 7 7 8 7 7 7 7 7 6 4 2 1 1 1 1 1

1 1 1 1 4 6 8 8 7 7 8 8 8 8 8 8 6 5 5 7 7 7 7 8 7 7 7 7 7 6 4 2 1 1 1 1

1 2 2 1 1 3 6 7 8 7 7 7 6 5 7 8 8 7 6 7 8 8 7 6 6 6 7 7 7 7 6 3 2 2 2 1

1 3 3 2 1 3 5 7 7 7 6 4 4 6 8 8 8 8 8 8 7 6 4 4 6 7 7 7 5 3 1 2 4 4 2

2 4 4 2 1 3 5 6 8 6 4 4 6 7 8 8 8 7 7 7 6 4 4 6 7 6 5 4 3 2 2 3 3 2

1 2 4 4 2 1 3 5 7 7 6 6 7 7 7 8 8 7 6 6 7 6 6 7 7 5 3 3 4 4 3 2 2 1

1 2 4 4 2 1 1 3 5 7 8 7 7 7 8 8 8 7 7 6 7 7 8 7 5 3 1 2 4 4 2

2 4 4 2 1 1 1 3 5 7 7 7 8 8 8 8 8 8 7 7 7 7 5 3 1 1 2 3 2 1 1 1

1 3 3 2 1 1 1 3 5 7 7 7 8 8 8 8 8 7 7 6 6 3 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 3 5 7 7 7 8 8 8 8 8 7 6 3 2 1 1 1 1 3 2 1 1 1

1 1 1 1 1 1 1 1 1 3 5 7 7 7 8 8 8 7 5 3 2 1 1 1 1 1 2 4 4 2 1 1 1

1 2 3 2 2 1 1 1 1 1 3 5 7 7 8 8 7 5 3 1 1 1 1 2 4 4 2 1 1 1

1 2 4 4 3 2 2 1 1 1 1 3 5 7 8 7 5 3 1 1 1 1 1 2 2 2 1 1 1 1 1

2 4 3 2 1 1 1 1 1 1 1 1 3 5 6 5 3 1 1 1

1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 3 4 3 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1

AVG3X3Out2\_thr\_7\_PP:

31 40 0 1

1 1 1

1 1 1

1 1 1

1 1 1 1 1

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1

1 1 1

AVG3X3Out2\_thr\_8\_PP:

31 40 0 1

1

1

1

1

1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

1 1 1 1 1

1 1 1 1 1 1

1 1 1 1

1 1

1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1

1 1 1 1 1

1 1 1

1 1

1