CSC 381 Computer Vision ( Java )

Project 4.1: Implementation of the four basic Morphology Operation

Han Wen Loh

Soft copy: 03/10/2019

Hard copy: 03/14/2019

Algorithm Steps:

step 0: open all files

( numRowsImg, numColsImg, minImg, maxImg ) 🡨 get from input1

( numRowsStrctElem, numColsStrctElem, minStrctElem, maxStrctElem ) 🡨 get from input2

( rowOrigin, colOrigin) 🡨 get from input2

step 1: computeFrameSize ( )

step 2: - dynamically allocate imgAry with extra rows and extra columns

- loadImage // load input file to imgAry, begins at (rowFrameSize, colFrameSize) and ends at ??

- zeroFrameImg () // in Java, you do not need to do this, but if in C++, you will need to do this

- prettyPrint (imgAry) // pretty print imgAry to the \*console\* of "Input Image"

- dynamically allocate morphAry with extra rows and extra columns

step 3:

- dynamically allocate structElemAry

- loadstruct // load input2 file to structElem array

- prettyPrint (structElemAry)// pretty print to the \*console\* of "Structuring Element"

step 4: - initMorphAry( ) // initialize morphAry to zero

- call dilation // see your lecture note

- prettyPrint (morphAry) // pretty print to the \*console\* the result of dilation

- outputResult //write the delation result to Output1 (argv[3])

step 5: - initMorphAry( ) // initialize morphAry to zero

- call erosion // see your lecture note

- prettyPrint (morphAry) // pretty print to the \*console\* the result of erosion

- outputResult //write the delation result to Output2 (argv[4])

step 6: - initMorphAry( ) // initialize morphAry to zero

- call closing // By call those two morphological ops one after the other

- prettyPrint (morphAry) // pretty print to the \*console\* the result of closing

- outputResult //write the closing result to Output3 (argv[5])

step 7: - initMorphAry( ) // initialize morphAry to zero

- call opening // By call those two morphological ops one after the other

- prettyPrint (morphAry) // pretty print to the \*console\* with "Opening Result"

- outputResult //write the opening result to Output4 (argv[6])

step 8: close all files

import java.io.PrintWriter;

import java.util.Scanner;

public class BasicMorphology {

private int numRowsImg, numColsImg, minImg, maxImg;

private int numRowsStructEle, numColsStructEle, minStructEle, maxStructEle;

private int rowOrigin, colOrigon;

private int rowFrameSize, colFrameSize;

private int[][] imgAry;

private int[][] structEleAry;

private int[][] morphAry;

private int[][] tempAry;

public BasicMorphology(Scanner img, Scanner structEle) {

numRowsImg = img.nextInt();

numColsImg = img.nextInt();

minImg = img.nextInt();

maxImg = img.nextInt();

numRowsStructEle = structEle.nextInt();

numColsStructEle = structEle.nextInt();

minStructEle = structEle.nextInt();

maxStructEle = structEle.nextInt();

rowOrigin = structEle.nextInt();

colOrigon = structEle.nextInt();

rowFrameSize = rowOrigin;

colFrameSize = colOrigon;

imgAry = new int[numRowsImg + rowFrameSize \* 2][numColsImg + colFrameSize \* 2];

structEleAry = new int[numRowsStructEle][numColsStructEle];

morphAry = new int[numRowsImg + rowFrameSize \* 2][numColsImg + colFrameSize \* 2];

tempAry = new int[numRowsImg + rowFrameSize \* 2][numColsImg + colFrameSize \* 2];

}

public void loadImage(Scanner img, String s) {

for (int i = rowFrameSize; i < numRowsImg + rowFrameSize; ++i) {

for (int j = colFrameSize; j < numColsImg + colFrameSize; ++j) {

imgAry[i][j] = img.nextInt();

}

}

prettyPrint(imgAry, s);

}

private void prettyPrint(int[][] ary, String s) {

System.out.format("%s\n", s);

for (int[] i : ary) {

for (int j : i) {

if (j > 0)

System.out.format("1 ");

else

System.out.format(" ");

}

System.out.format("\n");

}

}

private void outputPrint(int[][] ary, PrintWriter outFile) {

outFile.format("%d %d %d %d\n", numRowsImg, numColsImg, minImg, maxImg);

for (int m = rowFrameSize; m < numRowsImg + rowFrameSize; ++m) {

for (int n = colFrameSize; n < numColsImg + colFrameSize; ++n) {

if (ary[m][n] > 0)

outFile.format("1 ");

else

outFile.format("0 ");

}

outFile.format("\n");

}

}

public void loadStruct(Scanner structEle, String s) {

for (int i = 0; i < numRowsStructEle; ++i) {

for (int j = 0; j < numColsStructEle; ++j) {

structEleAry[i][j] = structEle.nextInt();

}

}

prettyPrint(structEleAry, s);

}

private void initAry(int[][] ary) {

for (int i = 0; i < ary.length; ++i) {

for (int j = 0; j < ary[i].length; ++j)

ary[i][j] = 0;

}

}

private void dilation(int[][] image, int[][] processedImg) {

int iOffset = 0, jOffset = 0;

for (int m = rowFrameSize; m < numRowsImg + rowFrameSize; ++m) {

for (int n = colFrameSize; n < numColsImg + colFrameSize; ++n) {

if (image[m][n] >= 1) {

iOffset = m + (0 - rowOrigin);

jOffset = n + (0 - colOrigon);

for (int r = 0; r < numRowsStructEle; ++r)

for (int c = 0; c < numColsStructEle; ++c)

if (structEleAry[r][c] >= 1)

processedImg[iOffset + r][jOffset + c] = 1;

}

}

}

}

private void erosion(int[][] image, int[][] processedImg) {

int iOffset = 0, jOffset = 0;

for (int m = rowFrameSize; m < numRowsImg + rowFrameSize; ++m) {

for (int n = colFrameSize; n < numColsImg + colFrameSize; ++n) {

if (image[m][n] >= 1) {

iOffset = m + (0 - rowOrigin);

jOffset = n + (0 - colOrigon);

out: {

for (int r = 0; r < numRowsStructEle; ++r)

for (int c = 0; c < numColsStructEle; ++c)

if (structEleAry[r][c] >= 1 && image[iOffset + r][jOffset + c] <= 0)

break out;

processedImg[m][n] = 1;

}

}

}

}

}

public void dilationOp(PrintWriter outFile, String fname) {

initAry(morphAry);

dilation(imgAry, morphAry);

prettyPrint(morphAry, "Dilation Operation");

outputPrint(morphAry, outFile);

}

public void erosionOp(PrintWriter outFile, String fname) {

initAry(morphAry);

erosion(imgAry, morphAry);

prettyPrint(morphAry, "Erosion Operation");

outputPrint(morphAry, outFile);

}

public void closingOp(PrintWriter outFile, String fname) {

initAry(morphAry);

initAry(tempAry);

dilation(imgAry, tempAry);

erosion(tempAry, morphAry);

prettyPrint(morphAry, "Closing Operation");

outputPrint(morphAry, outFile);

}

public void openingOp(PrintWriter outFile, String fname) {

initAry(morphAry);

initAry(tempAry);

erosion(imgAry, tempAry);

dilation(tempAry, morphAry);

prettyPrint(morphAry, "Opening Operation");

outputPrint(morphAry, outFile);

}

}

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner inFile1, inFile2;

PrintWriter outFile1, outFile2, outFile3, outFile4;

BasicMorphology basicMorphology;

try {

inFile1 = new Scanner(new FileReader(args[0]));

inFile2 = new Scanner(new FileReader(args[1]));

outFile1 = new PrintWriter(new FileOutputStream(args[2]));

outFile2 = new PrintWriter(new FileOutputStream(args[3]));

outFile3 = new PrintWriter(new FileOutputStream(args[4]));

outFile4 = new PrintWriter(new FileOutputStream(args[5]));

basicMorphology = new BasicMorphology(inFile1, inFile2);

basicMorphology.loadImage(inFile1, args[0]);

basicMorphology.loadStruct(inFile2, args[1]);

basicMorphology.dilationOp(outFile1, args[2]);

basicMorphology.erosionOp(outFile2, args[3]);

basicMorphology.closingOp(outFile3, args[4]);

basicMorphology.openingOp(outFile4, args[5]);

inFile1.close();

inFile2.close();

outFile1.close();

outFile2.close();

outFile3.close();

outFile4.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

Morphology\_Img1.txt

1 1

1 1 1 1

1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1

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

1 1 1 1 1 1 1 1 1 1 1

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

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

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

1 1 1 1 1 1 1 1 1 1 1

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

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

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1

1 1 1

1

1 1

1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

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

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

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

1 1 1 1 1 1 1 1 1 1 1

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

1 1 1 1 1 1 1 1 1 1

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

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

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

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

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1

1 1 1

Morphology\_StrucElem1.txt

1

1 1 1

1

Dilation Operation

1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

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

1 1 1 1 1 1 1 1 1 1 1

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1 1 1 1 1 1 1 1 1 1

1 1 1

Erosion Operation

1

1 1 1

1 1 1

1 1 1

1 1 1 1 1 1

1 1 1 1 1 1 1

1

1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1

1 1 1 1 1

1 1 1

1 1

1 1

1

1

1 1

1

1 1 1

1 1 1

1 1 1

1 1

1 1

1 1 1

1 1 1 1

1 1 1 1

1 1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1 1

1 1 1

1

1

Closing Operation

1 1

1 1 1 1

1 1 1

1 1 1 1 1 1 1

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

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

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

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

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

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

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

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

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

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

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

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1

1 1 1

1

1 1

1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1

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

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

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

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

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

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

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

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

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

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

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1

Opening Operation

1

1 1 1

1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

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

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

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

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

1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1

1 1 1 1

1 1 1

1

1

1 1 1 1

1 1 1 1 1 1

1 1 1 1

1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1

1 1 1

1 1 1

1

Morphology\_Img2.txt

1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1

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

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

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

1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1

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

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

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

1 1 1 1 1 1

1 1 1 1 1 1 1 1

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

1 1 1 1 1 1 1

1 1 1 1 1 1

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

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

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

Morphology\_StrucElem2.txt

1 1 1

1 1 1

1 1 1

Dilation Operation

1 1 1 1 1 1 1 1 1 1

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

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

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

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

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

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

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

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

1 1 1 1 1 1 1 1 1 1

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

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

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

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

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

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

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

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

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

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

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

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

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

Erosion Operation

1 1

1 1

1 1

1 1

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

1 1

1 1

1 1

1 1

1 1

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

1 1

1 1

1 1

1 1

1 1

1 1

1 1

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

Closing Operation

1 1 1 1 1 1

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

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

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

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

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

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

1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1

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

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

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

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

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

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

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

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

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

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

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

Opening Operation

1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1

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

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

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

1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1

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

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

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

1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1

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

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

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