CSC 381-34: Proj3 (JAVA)

Swrajit Paul

Due date: Sept. 27, 2018

Algorithm steps:
III. Algorithms
****

step 0: read the image header
dynamically allocate zeroFramedAry and all other arrays

step 1: load the input image onto zeroFramedAry

step 2: - ConnectCC_Pass1 // as taught in class
- prettyprint the result of pass1// with proper caption
- print EQAry // with index up to newLable with proper caption

step 3: - ConnectCC_Pass2 // as taught in class
- prettyprint the result of pass2// with caption
- print EQAry // with index up to newLable with caption

step 4: - manageEQAry // see algorithm below.
- print EQAry // with index up to newLable with caption

step 5: - ConnectCC_Pass3 // In the pass3, you will use the EQAry to relabel the components;
// keep track the newMin newMax
// as well as compute the property of each c.c.
// and store the c. c. properties

- prettyprint the result of pass3 of the connected c.c. // with caption
- Output the result of pass3 to outFile2 with updated image header
- print the propertis of the connected c.c. // with proper caption

****
Algoritm for manageEQAry
****
step 1: trueLabel <-- 0
step 2: index <-- 1
step 3: if EQAry[index] == index
trueLabel++
EQAry[index] <-- trueLabel
else
EQAry[index] <-- EQAry[EQAry[index]]

step 4: index++
step 5: repeat step 3 to 4 until index > newLabel

SOURCE CODE

```
/**
 * Project 3
 * Author: Swrajit Paul
 */

import java.io.*;
import java.util.Scanner;

class Property {

	int label;
	int numPixels;
	int minRow;
	int minCol;
	int maxRow;
	int maxCol;

	Property(){

	}
}

public class Project3 {

	static int numRows;
	static int numCols;
	static int minVal;
	static int maxVal;
	static int newMin;
	static int newMax;

	static int newLabel = 0;

	static int[][] zeroFramedAry;
	static int[] EQAry;
   static int[] neighborAry = new int[4];
   static Property[] cc;

   static FileInputStream fInput = null;
   static FileOutputStream fOutputone;
   static FileOutputStream fOutputtwo;
   static FileOutputStream fOutputthree;

   static Scanner inputfile;

   public Project3() {

   }

   public static void loadImage(int[][] arrayB) {

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

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

```
                        arrayB[i][j] = inputfile.nextInt();
                    }
                }

        }

        public static void zeroFrame(int[][] array) {
                // java initializes array with 0, therefore this is redundant
        }

        public static void loadNeighbors(int i, int j) {
                for(int k = 0; k < neighborAry.length; k++) {
                        neighborAry[k] = 0;
                }

                neighborAry[0] = zeroFramedAry[i-1][j];
                neighborAry[1] = zeroFramedAry[i][j-1];
                neighborAry[2] = zeroFramedAry[i+1][j];
                neighborAry[3] = zeroFramedAry[i][j+1];
        }

        public static void ConnectCC_Pass1() {

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

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

                                if (zeroFramedAry[i][j] > 0) {

                                        loadNeighbors(i,j);

                                        // Case 1
                                        if (neighborAry[0] == 0 && neighborAry[1] == 0){
                                                zeroFramedAry[i][j] = ++newLabel;
                                        }

                                        // Case 2
                                        else if (neighborAry[0] != 0 && neighborAry[1] != 0 && neighborAry[0]
== neighborAry[1]){
                                                zeroFramedAry[i][j] = neighborAry[0];
                                        }

                                        // Case 3
                                        else if (neighborAry[0] != 0 || neighborAry[1] != 0){
                                                if (neighborAry[0] == 0 && neighborAry[1] != 0){
                                                        zeroFramedAry[i][j] = neighborAry[1];
                                                }
                                                if (neighborAry[0] != 0 && neighborAry[1] == 0){
                                                        zeroFramedAry[i][j] = neighborAry[0];
                                                }
                                                if (neighborAry[0] != 0 && neighborAry[1] != 0){
                                                        if (neighborAry[0] < neighborAry[1] ){
                                                                zeroFramedAry[i][j] = neighborAry[0];
                                                                updateEQAry(neighborAry[1],neighborAry[0]);
```

```
                                                }
                                                if (neighborAry[0] > neighborAry[1] ){
                                                        zeroFramedAry[i][j] = neighborAry[1];
                                                        updateEQAry(neighborAry[0], neighborAry[1]);
                                                }
                                        }
                                }
                        }
                }

        }

        public static void ConnectCC_Pass2() {

                for(int i = numRows+1; i>= 1; i--) {

                        for(int j = numCols+1; j >= 1; j--) {

                                if (zeroFramedAry[i][j] > 0){

                                        loadNeighbors(i,j);

                                        // Case 1
                                        // DO nothing

                                        // Case 2
                                        if ((neighborAry[2] != 0 && neighborAry[3] != 0) && (neighborAry[2] ==
neighborAry[3])){
                                                zeroFramedAry[i][j] = neighborAry[2];
                                        }

                                        // Case 3
                                        else if (neighborAry[2] != 0 || neighborAry[3] != 0){
                                                if (neighborAry[2] == 0 && neighborAry[3] != 0){
                                                        zeroFramedAry[i][j] = neighborAry[3];
                                                }
                                                if (neighborAry[2] != 0 && neighborAry[3] == 0){
                                                        zeroFramedAry[i][j] = neighborAry[2];
                                                }
                                                if (neighborAry[2] != 0 && neighborAry[3] != 0){
                                                        if (neighborAry[2] < neighborAry[3] ){
                                                                zeroFramedAry[i][j] = neighborAry[2];
                                                                updateEQAry(neighborAry[3],neighborAry[2]);
                                                        }
                                                        if (neighborAry[2] > neighborAry[3] ){
                                                                zeroFramedAry[i][j] = neighborAry[3];
                                                                updateEQAry(neighborAry[2],neighborAry[3]);
                                                        }
                                                }
                                        }
                                }
                        } // end of inner loop
                } // end of outer loop
```

```
}

public static void ConnectCC_Pass3() {

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

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

			if (zeroFramedAry[i][j] > 0) {

				zeroFramedAry[i][j] = EQAry[zeroFramedAry[i][j]];
			}
		}
	}

	newMin = 200000;
	newMax = 0;

	for(int i = 0; i <= newLabel; i++) {
		if(EQAry[i] > newMax){
			newMax = EQAry[i];
		}
		if (EQAry[i] < newMin){
			newMin = EQAry[i];
		}
	}

	PrintStream print = new PrintStream(fOutputtwo);
	print.println(numRows + " " + numCols + " " + newMin + " " + newMax + " ");
	for(int i = 1; i < numRows+1; i++) {

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

			if(zeroFramedAry[i][j] < 10){
				print.print(zeroFramedAry[i][j] + "  ");
			}
			else {
				print.print(zeroFramedAry[i][j] + " ");
			}

		}
		print.println();

	}
	cc = new Property[newMax+1];
	for(int k = 0; k < newMax+1; k++) {
		cc[k] = new Property();
	}

	for(int k = 1; k <= newMax; k++) {
		int countPixels = 0;
		int maxr, maxc, minc, minr;
		maxr = 0;
		maxc = 0;
		minc = numCols;
```

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

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

                                if(zeroFramedAry[i][j] == k){
                                        if(i-1 < minr){
                                                minr = i-1;
                                        }
                                        if(i-1 > maxr){
                                                maxr = i-1;
                                        }
                                        if(j-1 > maxc){
                                                maxc = j-1;
                                        }
                                        if(j-1 < minc){
                                                minc = j-1;
                                        }
                                        countPixels++;
                                }
                        }
                }


                cc[k].label = k;
                cc[k].numPixels = countPixels;
                cc[k].minRow = minr;
                cc[k].minCol = minc;
                cc[k].maxRow = maxr;
                cc[k].maxCol = maxc;

        }
}

public static void updateEQAry(int i, int j) {
        EQAry[i] = j;
}

public static void manageEQAry() {
        int trueLabel = 0;

        int index = 1;

        while(index <= newLabel) {

                if (EQAry[index] == index) {
                        trueLabel++;
                        EQAry[index] = trueLabel;
                }
                else
                        EQAry[index] = EQAry[EQAry[index]];

                index++;
        }
```

```
}

public static void printCCProperty() {

	PrintStream print = new PrintStream(fOutputthree);
	print.println(numRows + " " + numCols + " " + newMin + " " + newMax);
	print.println(newMax);
	for(int i = 1; i <= newMax; i++) {
		print.println(cc[i].label);
		print.println(cc[i].numPixels);
		print.println(cc[i].minRow + " " + cc[i].minCol);
		print.println(cc[i].maxRow + " " + cc[i].maxCol);
	}

}

public static void prettyPrint (int pass) {

	PrintStream print = new PrintStream(fOutputone);
	if(pass == 1 || pass == 2 || pass == 3) {
		print.println("This is the result of pass " + pass + ":");
		for(int i = 0; i < zeroFramedAry.length; i++) {

			for(int j = 0; j < zeroFramedAry[0].length; j++) {

				if(zeroFramedAry[i][j] > 0) {
					print.print(zeroFramedAry[i][j]);
				}
				else {
					print.print("  ");
				}
			}
			print.println();
		}

		print.println();
		print.println("This is the EQAry after pass " + pass);
		for(int i=0; i <= newLabel; i++) {
			print.println(i + " " + EQAry[i]);
		}
		print.println();
	}
	else {
		print.println("The EQAry after manageEQAry is:");
		for(int k = 0; k <= newLabel; k++){
			print.println(k + " "+ EQAry[k]);
		}
		print.println();
	}

}

public static void main(String[] args) {

	try {
```

```
                String inputone = args[0];
                String outputone = args[1];
                String outputtwo = args[2];
                String outputthree = args[3];

                fInput = new FileInputStream(inputone);
                fOutputone = new FileOutputStream(outputone);
                fOutputtwo = new FileOutputStream(outputtwo);
                fOutputthree = new FileOutputStream(outputthree);

            } catch (IOException e) {
                System.out.println("one of the arguments in missing or wrong");
            }


            inputfile = new Scanner(fInput);

            numRows = inputfile.nextInt();
            numCols = inputfile.nextInt();
            minVal = inputfile.nextInt();
            maxVal = inputfile.nextInt();

            zeroFramedAry = new int[numRows+2][numCols+2];;
            EQAry = new int[(numRows*numCols)/2];
            for(int i=0; i < EQAry.length; i++) {
                EQAry[i] = i;
            }

            loadImage(zeroFramedAry);
            ConnectCC_Pass1();
            prettyPrint(1);
            ConnectCC_Pass2();
            prettyPrint(2);
            manageEQAry();
            prettyPrint(4);
            ConnectCC_Pass3();
            prettyPrint(3);
            printCCProperty();


            inputfile.close();
            try {
                fInput.close();
            } catch (IOException e) {

                e.printStackTrace();
            }
        }

}
```

INPUT

Input Data 1

42 31 0 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 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0
0 0 1 1 0 0 1 1 1 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0
0 0 1 1 0 0 0 1 0 0 0 0 1 1 1 1 1 0 1 0 0 1 1 1 0 0 1 0 0 0 0
0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0
0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0
0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0
0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0
0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0
0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1 0 0
0 0 0 0 1 0 0 1 0 0 1 1 0 1 1 0 0 0 0 1 0 1 0 0 0 0 1 0 1 0 0
0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 0 1 0 1 0 0
0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 1 1 1 0 0 0 1 0 0 0 0 1 0 1 1 0
0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0
0 0 0 0 1 0 0 1 0 0 0 1 1 0 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 1 0
0 0 0 0 1 1 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 1 1 0
1 1 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0
1 1 1 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 0
0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0
0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 1 0 1 0 0
0 0 0 0 1 1 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 1 1 0 1 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 1 1 0 1 0 1 0 0
0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 1 0 0
0 0 0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0
0 0 0 1 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 0 0 0
0 0 0 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 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 1 0 0 1 0 0
0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 1 0 0 1 0 0
0 0 0 1 0 0 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 1 0 0
0 0 1 1 1 0 0 0 0 1 1 1 1 1 0 0 1 0 0 1 1 1 0 1 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 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 1 1 0 0 0 0 0 0 0
0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0
0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 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

Input Data 2:

42 31 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 0 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 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 1 1 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0
0 0 1 1 0 0 0 1 0 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 0 1 0 1 0 0 0 1 1 0 0 0 0
0 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0

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

OUTPUT

Output file 1 for Data 1:

This is the result of pass 1:

```
                              111
                                1                    22
                3              1          4      522
      66    733      8    1    9    4        2
      66      3        88811  9    444    2
          1010103              1      11  4      122
            10103            1311      14          2
          15    3          16              17        2  18
          15    3          16              17        2  18
          15    3    1919  16              17        2  1818
          15    3    1919  16              17        2  1818
          15    3    1919  16              17        2
          15    3    1919  16            2017        2  21
          15    3    1919  1616        22  17        2  21
          15    3    1919  16  23    24    17        2  21
          15    3    1919  16  232323      17        2  2121
```

```
         15    3    1919  16    23        17        2    21
         15    3      1919  252523    26          272    21
         1515  3      1919    25  2828  29              3021
  313131  15    3    3219            28  29              30
  313131  15        3332        34      352929    363636  30
            37    3833          34            3939    36  30
            37      3333      403434          3939  4136  30
          4237  434333        403434          3939393936  30
            3737            4440343434          3939  36  30
              37          45444034343434      4639    363630
            473737      484544403434343434              36
        4949      50  51484544403434343434            36
        4949        525148454440343434343434      53  36
                                                  53    54
                    55    5656    5757    5858      53    54
        59          55    5656    5757    5858    6053535353
      615959        5555555555    57    625858  63  53
          59        555555555555555555555555555
        645959      55555555555555555555555
        64  59        55      555555          65
        646459                                6565
      6664645959                          67    6565
        646459              686868      67        69
          64                686868    706767  717169
                              68  72727067676767
```

```
This is the EQAry after pass 1
0 0
1 1
2 2
3 3
4 4
5 2
6 6
7 3
8 1
9 9
10 3
11 11
12 2
13 1
14 14
15 15
16 16
17 17
18 18
19 19
20 17
21 21
22 22
23 23
24 24
25 23
26 26
27 2
28 28
29 29
30 21
```

```
31 31
32 19
33 32
34 34
35 29
36 30
37 37
38 33
39 36
40 34
41 39
42 37
43 33
44 40
45 44
46 39
47 37
48 45
49 49
50 50
51 48
52 51
53 53
54 53
55 55
56 55
57 55
58 55
59 59
60 53
61 59
62 55
63 55
64 59
65 65
66 64
67 67
68 68
69 69
70 67
71 67
72 70
```

This is the result of pass 2:

```
                            111
                              1                    22
              3               1          4      222
      66    333       1    1    9    4        2
      66      3         11111  9    444    2
          3333               1      11  4      22
            333             111      14          2
          15    3           16              17        2  18
          15    3           16              17        2  18
          15    3    1919   16              17        2  1818
          15    3    1919   16              17        2  1818
          15    3    1919   16              17        2
          15    3    1919   16            1717        2  21
          15    3    1919   1616        22  17        2  21
          15    3    1919   16  23    24    17        2  21
```

```
         15    3    1919  16  232323      17        2  2121
         15    3    1919  16    23        17        2    21
         15    3      1919  232323    26          22    21
         1515  3      1919    25  2828  29              2121
  313131  15    3    1919            28  29              30
  313131  15        3232        34      292929    303030  30
            37    3333          34            3030    30  30
            37      3333      343434          3030  3030  30
         3737  333333        343434          3030303030  30
           3737            3434343434          3939  30  30
             37          34343434343434      3939    303030
           373737      3434343434343434              36
       4949      50  34343434343434343434            36
       4949        343434343434343434343434      53  36
                                                 53    53
                   55    5555    5557    5555      53    53
       59          55    5555    5557    5555    5353535353
     595959        5555555555    55    555555  55  53
         59        55555555555555555555555555555
       595959      555555555555555555555555
       59  59        55      555555          65
       595959                                6565
     5959595959                          67    6565
       595959              686868      67        69
         64                686868    676767  676969
                             68  67676767676767


This is the EQAry after pass 2
0 0
1 1
2 2
3 3
4 4
5 2
6 6
7 3
8 1
9 9
10 3
11 11
12 2
13 1
14 14
15 15
16 16
17 17
18 18
19 19
20 17
21 21
22 22
23 23
24 24
25 23
26 26
27 2
28 28
29 29
```

```
30 21
31 31
32 19
33 32
34 34
35 29
36 30
37 37
38 33
39 30
40 34
41 39
42 37
43 33
44 40
45 44
46 39
47 37
48 45
49 49
50 50
51 48
52 51
53 53
54 53
55 55
56 55
57 55
58 55
59 59
60 53
61 59
62 55
63 55
64 59
65 65
66 64
67 67
68 68
69 67
70 67
71 67
72 70

The EQAry after manageEQAry is:
0 0
1 1
2 2
3 3
4 4
5 2
6 5
7 3
8 1
9 6
10 3
11 7
12 2
13 1
14 8
```

15 9
16 10
17 11
18 12
19 13
20 11
21 14
22 15
23 16
24 17
25 16
26 18
27 2
28 19
29 20
30 14
31 21
32 13
33 13
34 22
35 20
36 14
37 23
38 13
39 14
40 22
41 14
42 23
43 13
44 22
45 22
46 14
47 23
48 22
49 24
50 25
51 22
52 22
53 26
54 26
55 27
56 27
57 27
58 27
59 28
60 26
61 28
62 27
63 27
64 28
65 29
66 28
67 30
68 31
69 30
70 30
71 30
72 30

This is the result of pass 3:

```
                          111
                            1                    22
              3              1          4      222
      55    333      1    1    6    4        2
      55      3        11111  6    444    2
          3333              1      7  4      22
            333            111      8          2
          9    3          10              11        2  12
          9    3          10              11        2  12
          9    3    1313  10              11        2  1212
          9    3    1313  10              11        2  1212
          9    3    1313  10              11        2
          9    3    1313  10            1111        2  14
          9    3    1313  1010        15  11        2  14
          9    3    1313  10  16    17    11        2  14
          9    3    1313  10  161616      11        2  1414
          9    3    1313  10    16        11        2    14
          9    3      1313  161616    18          22    14
          99  3      1313    16  1919  20              1414
  212121  9    3    1313            19  20              14
  212121  9        1313        22      202020    141414  14
            23    1313          22            1414    14  14
            23      1313      222222          1414  1414  14
          2323  131313        222222          1414141414  14
            2323            2222222222          1414  14  14
              23          22222222222222      1414    141414
            232323      222222222222222222              14
        2424      25  2222222222222222222222            14
        2424        22222222222222222222222222      26  14
                                                    26    26
                    27    2727    2727    2727      26    26
        28          27    2727    2727    2727    2626262626
      282828        2727272727    27    272727  27  26
          28        272727272727272727272727272727
        282828      27272727272727272727272727
        28  28        27      272727          29
        282828                                2929
      2828282828                            30    2929
        282828                313131      30        30
          28                  313131    303030  303030
                                31  30303030303030
```

```
This is the EQAry after pass 3
0 0
1 1
2 2
3 3
4 4
5 2
6 5
7 3
8 1
9 6
10 3
11 7
12 2
13 1
```

14 8
15 9
16 10
17 11
18 12
19 13
20 11
21 14
22 15
23 16
24 17
25 16
26 18
27 2
28 19
29 20
30 14
31 21
32 13
33 13
34 22
35 20
36 14
37 23
38 13
39 14
40 22
41 14
42 23
43 13
44 22
45 22
46 14
47 23
48 22
49 24
50 25
51 22
52 22
53 26
54 26
55 27
56 27
57 27
58 27
59 28
60 26
61 28
62 27
63 27
64 28
65 29
66 28
67 30
68 31
69 30
70 30
71 30
72 30

Output file 2 for Data 1:

```
42 31 0 31
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  1  0  0  0  0  0  0  0  0  0  0  2  2  0  0  0
0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  1  0  0  0  0  0  4  0  0  0  2  2  2  0  0  0
0  0  5  5  0  0  3  3  3  0  0  0  1  0  0  1  0  0  6  0  0  4  0  0  0  0  2  0  0  0  0
0  0  5  5  0  0  0  3  0  0  0  0  1  1  1  1  1  0  6  0  0  4  4  4  0  0  2  0  0  0  0
0  0  0  0  3  3  3  3  0  0  0  0  0  0  0  1  0  0  0  7  0  4  0  0  0  2  2  0  0  0  0
0  0  0  0  0  3  3  3  0  0  0  0  0  0  1  1  1  0  0  0  8  0  0  0  0  0  2  0  0  0  0
0  0  0  0  9  0  0  3  0  0  0  0  0  10 0  0  0  0  0  0  0  0  11 0  0  0  0  2  0  12 0  0
0  0  0  0  9  0  0  3  0  0  0  0  0  10 0  0  0  0  0  0  0  0  11 0  0  0  0  2  0  12 0  0
0  0  0  0  9  0  0  3  0  0  13 13 0  10 0  0  0  0  0  0  0  0  11 0  0  0  0  2  0  12 12 0
0  0  0  0  9  0  0  3  0  0  13 13 0  10 0  0  0  0  0  0  0  0  11 0  0  0  0  2  0  12 12 0
0  0  0  0  9  0  0  3  0  0  13 13 0  10 0  0  0  0  0  0  0  0  11 0  0  0  0  2  0  0  0  0
0  0  0  0  9  0  0  3  0  0  13 13 0  10 0  0  0  0  0  0  0  11 11 0  0  0  0  2  0  14 0  0
0  0  0  0  9  0  0  3  0  0  13 13 0  10 10 0  0  0  0  0  15 0  11 0  0  0  0  2  0  14 0  0
0  0  0  0  9  0  0  3  0  0  13 13 0  10 0  16 0  0  17 0  0  11 0  0  0  0  2  0  14 0  0
0  0  0  0  9  0  0  3  0  0  13 13 0  10 0  16 16 16 0  0  0  11 0  0  0  0  2  0  14 14 0
0  0  0  0  9  0  0  3  0  0  13 13 0  10 0  0  16 0  0  0  0  11 0  0  0  0  2  0  0  14 0
0  0  0  0  9  0  0  3  0  0  0  13 13 0  16 16 16 0  0  18 0  0  0  0  0  2  2  0  0  14 0
0  0  0  0  9  9  0  3  0  0  0  13 13 0  0  16 0  19 19 0  20 0  0  0  0  0  0  0  14 14 0
21 21 21 0  9  0  0  3  0  0  13 13 0  0  0  0  0  0  19 0  20 0  0  0  0  0  0  0  14 0  0
21 21 21 0  9  0  0  0  0  13 13 0  0  0  0  22 0  0  0  20 20 20 0  0  14 14 14 0  14 0  0
0  0  0  0  0  23 0  0  13 13 0  0  0  0  0  22 0  0  0  0  0  0  14 14 0  0  14 0  14 0  0
0  0  0  0  0  23 0  0  0  13 13 0  0  0  22 22 22 0  0  0  0  0  14 14 0  14 14 0  14 0  0
0  0  0  0  23 23 0  13 13 13 0  0  0  0  22 22 22 0  0  0  0  0  14 14 14 14 14 0  14 0  0
0  0  0  0  0  23 23 0  0  0  0  0  0  22 22 22 22 22 0  0  0  0  0  14 14 0  14 0  14 0  0
0  0  0  0  0  0  23 0  0  0  0  0  22 22 22 22 22 22 22 0  0  0  14 14 0  0  14 14 14 0  0
0  0  0  0  0  23 23 23 0  0  0  22 22 22 22 22 22 22 22 22 0  0  0  0  0  0  0  14 0  0  0
0  0  0  24 24 0  0  0  25 0  22 22 22 22 22 22 22 22 22 22 22 0  0  0  0  0  0  14 0  0  0
0  0  0  24 24 0  0  0  0  22 22 22 22 22 22 22 22 22 22 22 22 22 0  0  0  26 0  14 0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  26 0  0  26 0  0
0  0  0  0  0  0  0  0  0  27 0  0  27 27 0  0  27 27 0  0  27 27 0  0  0  26 0  0  26 0  0
0  0  0  28 0  0  0  0  0  27 0  0  27 27 0  0  27 27 0  0  27 27 0  0  26 26 26 26 26 0  0
0  0  28 28 28 0  0  0  0  27 27 27 27 27 0  0  27 0  0  27 27 27 0  27 0  26 0  0  0  0  0
0  0  0  0  28 0  0  0  0  27 27 27 27 27 27 27 27 27 27 27 27 27 27 0  0  0  0  0  0  0
0  0  0  28 28 28 0  0  0  27 27 27 27 27 27 27 27 27 27 27 27 27 0  0  0  0  0  0  0  0
0  0  0  28 0  28 0  0  0  0  27 0  0  0  27 27 27 0  0  0  0  0  29 0  0  0  0  0  0  0  0
0  0  0  28 28 28 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  29 29 0  0  0  0  0  0
0  0  28 28 28 28 28 0  0  0  0  0  0  0  0  0  0  0  0  0  0  30 0  0  29 29 0  0  0  0  0
0  0  0  28 28 28 0  0  0  0  0  0  0  0  31 31 31 0  0  0  30 0  0  0  0  30 0  0  0  0  0
0  0  0  0  28 0  0  0  0  0  0  0  0  0  31 31 31 0  0  30 30 30 0  30 30 30 0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  31 0  30 30 30 30 30 30 30 0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Output file 3 for Data 1:

```
42 31 0 31
31
1
16
0 12
6 16
2
22
1 25
17 27
3
25
```

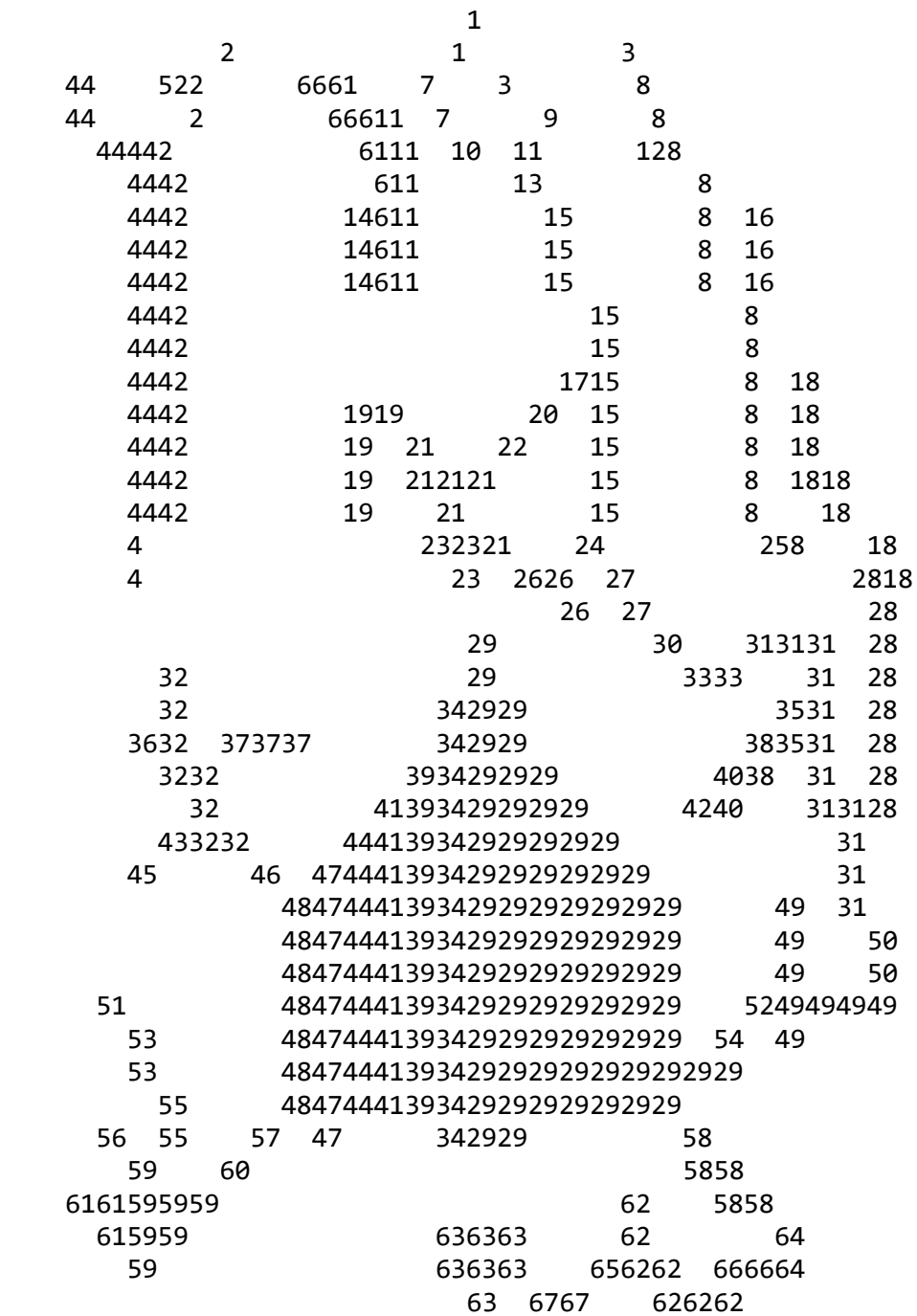
```
2 4
19 8
4
6
2 21
5 23
5
4
3 2
4 3
6
2
3 18
4 18
7
1
5 19
5 19
8
1
6 20
6 20
9
15
7 4
20 5
10
11
7 13
16 14
11
11
7 20
16 21
12
6
7 28
10 29
13
31
9 7
23 12
14
41
12 22
28 29
15
1
13 19
13 19
16
9
14 14
18 17
17
1
14 18
14 18
18
1
```

```
17 19
17 19
19
3
18 17
19 18
20
5
18 19
20 21
21
6
19 0
20 2
22
53
20 9
28 21
23
10
21 4
26 7
24
4
27 3
28 4
25
1
27 8
27 8
26
11
28 24
32 28
27
56
30 9
35 23
28
22
31 2
39 6
29
5
35 22
37 24
30
16
37 17
40 25
31
7
38 14
40 16

----------------------------ABOVE OUTPUTS IS FOR DATA #1------------------------------
```

Output file 1 for Data 2:

This is the result of pass 1:



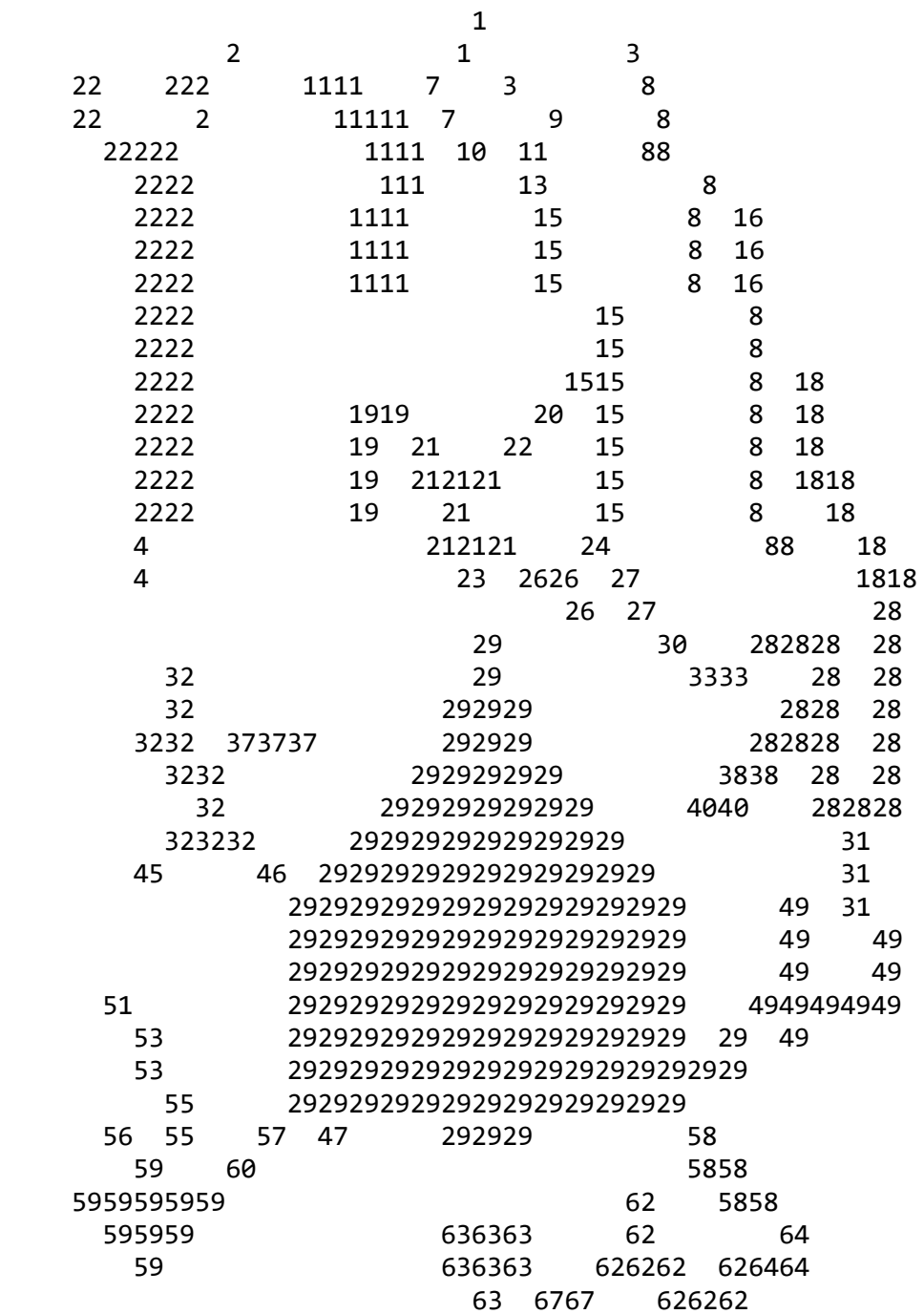
This is the EQAry after pass 1:

0 0
1 1
2 2
3 3
4 2
5 2
6 1
7 7
8 8

9 9
10 10
11 11
12 8
13 13
14 6
15 15
16 16
17 15
18 18
19 19
20 20
21 21
22 22
23 21
24 24
25 8
26 26
27 27
28 18
29 29
30 30
31 28
32 32
33 33
34 29
35 31
36 32
37 37
38 35
39 34
40 38
41 39
42 40
43 32
44 41
45 45
46 46
47 44
48 47
49 49
50 49
51 51
52 49
53 53
54 29
55 55
56 56
57 57
58 58
59 59
60 60
61 59
62 62

63 63
64 64
65 62
66 62
67 67

This is the result of pass 2:



This is the EQAry after pass 2:
0 0
1 1
2 2

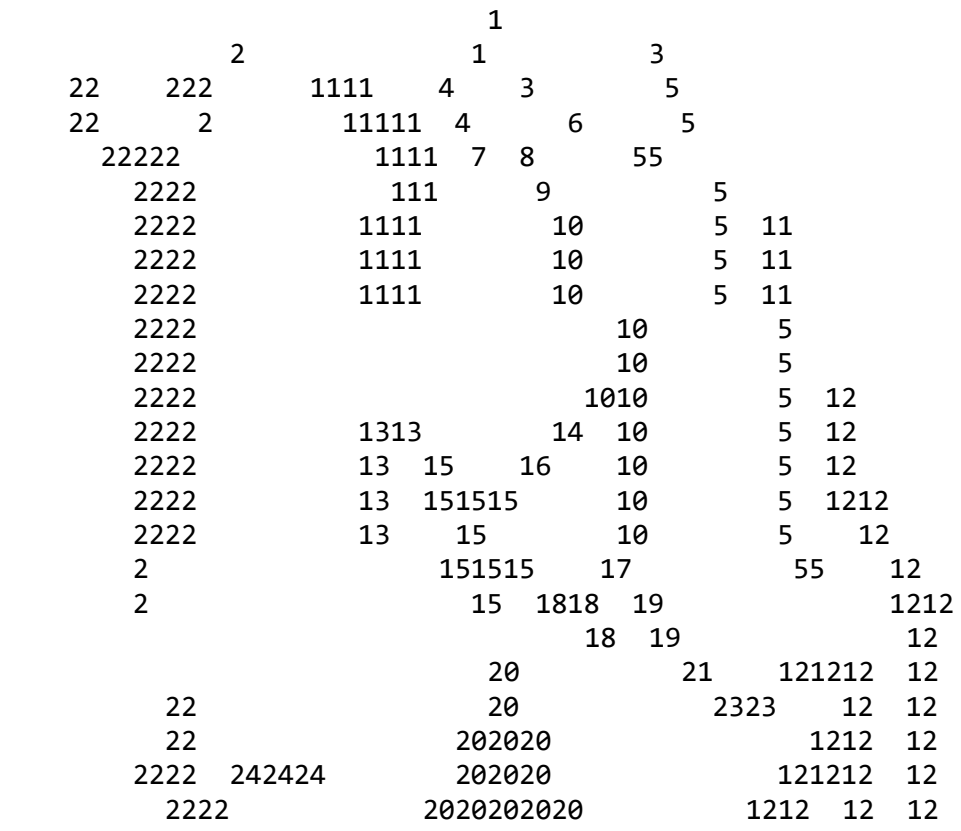
3 3
4 2
5 2
6 1
7 7
8 8
9 9
10 10
11 11
12 8
13 13
14 6
15 15
16 16
17 15
18 18
19 19
20 20
21 21
22 22
23 21
24 24
25 8
26 26
27 27
28 18
29 29
30 30
31 28
32 32
33 33
34 29
35 31
36 32
37 37
38 28
39 34
40 38
41 39
42 40
43 32
44 41
45 45
46 46
47 29
48 47
49 49
50 49
51 51
52 49
53 53
54 29
55 55
56 56

```
57 57
58 58
59 59
60 60
61 59
62 62
63 63
64 62
65 62
66 62
67 67

The EQAry after manageEQAry is:
0 0
1 1
2 2
3 3
4 2
5 2
6 1
7 4
8 5
9 6
10 7
11 8
12 5
13 9
14 1
15 10
16 11
17 10
18 12
19 13
20 14
21 15
22 16
23 15
24 17
25 5
26 18
27 19
28 12
29 20
30 21
31 12
32 22
33 23
34 20
35 12
36 22
37 24
38 12
39 20
40 12
```

41 20
42 12
43 22
44 20
45 25
46 26
47 20
48 20
49 27
50 27
51 28
52 27
53 29
54 20
55 30
56 31
57 32
58 33
59 34
60 35
61 34
62 36
63 37
64 36
65 36
66 36
67 38

This is the result of pass 3:



```
           22          20202020202020      1212    121212
         222222      202020202020202020              12
       25      26  2020202020202020202020            12
                 20202020202020202020202020      27  12
                 20202020202020202020202020      27    27
                 20202020202020202020202020      27    27
     28          20202020202020202020202020    2727272727
       29        20202020202020202020202020  20  27
       29        202020202020202020202020202020
         30      20202020202020202020202020
     31  30    32  20      202020          33
       34    35                            3333
   3434343434                          36    3333
     343434               373737      36        36
       34                 373737    363636  363636
                            37  3838    363636
```

This is the EQAry after pass 3:

```
0 0
1 1
2 2
3 3
4 2
5 2
6 1
7 4
8 5
9 6
10 7
11 8
12 5
13 9
14 1
15 10
16 11
17 10
18 12
19 13
20 14
21 15
22 16
23 15
24 17
25 5
26 18
27 19
28 12
29 20
30 21
31 12
32 22
33 23
34 20
```

35 12
36 22
37 24
38 12
39 20
40 12
41 20
42 12
43 22
44 20
45 25
46 26
47 20
48 20
49 27
50 27
51 28
52 27
53 29
54 20
55 30
56 31
57 32
58 33
59 34
60 35
61 34
62 36
63 37
64 36
65 36
66 36
67 38

Output file 2 for Data 2:

42 31 0 38

```
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  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  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  1  0  0  0  0  0  3  0  0  0  0  0  0  0  0
0  0  2  2  0  0  2  2  2  0  0  0  1  1  1  1  0  0  4  0  0  3  0  0  0  0  5  0  0  0
0  0  2  2  0  0  0  2  0  0  0  0  1  1  1  1  1  0  4  0  0  0  6  0  0  0  5  0  0  0
0  0  0  2  2  2  2  2  0  0  0  0  0  0  1  1  1  1  0  7  0  8  0  0  0  5  5  0  0  0
0  0  0  0  2  2  2  2  0  0  0  0  0  0  1  1  1  0  0  0  9  0  0  0  0  0  5  0  0  0
0  0  0  0  2  2  2  2  0  0  0  0  0  1  1  1  1  0  0  0  0  10 0  0  0  0  5  0  11 0
0  0  0  0  2  2  2  2  0  0  0  0  0  1  1  1  1  0  0  0  0  10 0  0  0  0  5  0  11 0
0  0  0  0  2  2  2  2  0  0  0  0  0  1  1  1  1  0  0  0  0  10 0  0  0  0  5  0  11 0
0  0  0  0  2  2  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  10 0  0  0  0  5  0  0  0
0  0  0  0  2  2  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  10 0  0  0  0  5  0  0  0
0  0  0  0  2  2  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  10 10 0  0  0  5  0  12 0
0  0  0  0  2  2  2  2  0  0  0  0  0  13 13 0  0  0  0  14 0  10 0  0  0  0  5  0  12 0
0  0  0  0  2  2  2  2  0  0  0  0  0  13 0  15 0  0  16 0  0  10 0  0  0  0  5  0  12 0
0  0  0  0  2  2  2  2  0  0  0  0  0  13 0  15 15 15 0  0  0  10 0  0  0  0  5  0  12 12 0
0  0  0  0  2  2  2  2  0  0  0  0  0  13 0  0  15 0  0  0  0  10 0  0  0  0  5  0  0  12 0
0  0  0  0  2  0  0  0  0  0  0  0  0  0  15 15 15 0  0  17 0  0  0  0  0  5  5  0  0  12 0
0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  15 0  18 18 0  19 0  0  0  0  0  0  0  12 12 0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  18 0  19 0  0  0  0  0  0  0  12 0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  20 0  0  0  0  0  0  21 0  0  12 12 12 0  12 0  0
0  0  0  0  0  22 0  0  0  0  0  0  0  0  0  0  20 0  0  0  0  0  0  23 23 0  0  12 0  12 0  0
0  0  0  0  0  22 0  0  0  0  0  0  0  0  20 20 20 0  0  0  0  0  0  0  0  12 12 0  12 0  0
0  0  0  0  22 22 0  24 24 24 0  0  0  0  20 20 20 0  0  0  0  0  0  0  12 12 12 0  12 0  0
0  0  0  0  0  22 22 0  0  0  0  0  0  20 20 20 20 20 0  0  0  0  0  12 12 0  12 0  12 0  0
0  0  0  0  0  0  22 0  0  0  0  0  20 20 20 20 20 20 0  0  0  12 12 0  0  12 12 12 0  0
```

```
0  0  0  0  0  22 22 22 0  0  0  20 20 20 20 20 20 20 20 20 0  0  0  0  0  0  0  12 0  0  0
0  0  0  0  25 0  0  0  26 0  20 20 20 20 20 20 20 20 20 20 20 0  0  0  0  0  0  12 0  0  0
0  0  0  0  0  0  0  0  0  20 20 20 20 20 20 20 20 20 20 20 20 0  0  0  27 0  12 0  0  0
0  0  0  0  0  0  0  0  0  20 20 20 20 20 20 20 20 20 20 20 20 20 0  0  0  27 0  0  27 0  0
0  0  0  0  0  0  0  0  0  20 20 20 20 20 20 20 20 20 20 20 20 20 0  0  0  27 0  0  27 0  0
0  0  0  28 0  0  0  0  0  20 20 20 20 20 20 20 20 20 20 20 20 20 0  0  27 27 27 27 27 0  0
0  0  0  0  29 0  0  0  0  20 20 20 20 20 20 20 20 20 20 20 20 20 0  20 0  27 0  0  0  0  0
0  0  0  0  29 0  0  0  0  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 0  0  0  0  0  0  0
0  0  0  0  0  30 0  0  0  20 20 20 20 20 20 20 20 20 20 20 20 20 0  0  0  0  0  0  0  0  0
0  0  0  31 0  30 0  0  32 0  20 0  0  0  20 20 20 0  0  0  0  0  33 0  0  0  0  0  0  0  0
0  0  0  0  34 0  0  35 0  0  0  0  0  0  0  0  0  0  0  0  0  0  33 33 0  0  0  0  0  0  0
0  0  34 34 34 34 34 0  0  0  0  0  0  0  0  0  0  0  0  0  0  36 0  0  33 33 0  0  0  0  0  0
0  0  0  34 34 34 0  0  0  0  0  0  0  0  37 37 37 0  0  0  36 0  0  0  0  36 0  0  0  0  0  0
0  0  0  0  34 0  0  0  0  0  0  0  0  0  37 37 37 0  0  36 36 36 0  36 36 36 0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  37 0  38 38 0  0  36 36 36 0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Output file 3 for Data 2:

```
42 31 0 38
38
1
30
1 12
9 17
2
60
2 2
18 8
3
2
2 21
3 21
4
2
3 18
4 18
5
17
3 25
17 26
6
1
4 22
4 22
7
1
5 19
5 19
8
1
5 21
5 21
9
1
6 20
6 20
10
11
7 20
16 21
```

```
11
3
7 28
9 28
12
35
12 22
28 29
13
5
13 13
16 14
14
1
13 19
13 19
15
9
14 14
18 17
16
1
14 18
14 18
17
1
17 19
17 19
18
3
18 17
19 18
19
2
18 20
19 20
20
138
20 9
35 23
21
1
20 21
20 21
22
10
21 4
26 7
23
2
21 22
21 23
24
3
```

```
23 7
23 9
25
1
27 4
27 4
26
1
27 8
27 8
27
11
28 24
32 28
28
1
31 3
31 3
29
2
32 4
33 4
30
2
34 5
35 5
31
1
35 3
35 3
32
1
35 8
35 8
33
5
35 22
37 24
34
10
36 2
39 6
35
1
36 7
36 7
36
12
37 19
40 25
37
7
38 14
40 16
```

```
38
2
40 17
40 18
```