CV
Project 5: Connected Components

Adrian Noa
Due 10/29/2022

My Algorithms:

connectPass1(int connectness):
This method takes the connectness input, 4 or 8, and performs the first pass of the connected component. My algorithm uses an array list and 2 case methods. 1 and 3. Case 2 is excluded because it is essentially case 3 without updating the equivalency table.
If there is one nonzero neighbor, case 2 applies. If there are 2 or more non zero neighbor case 2 or 3 apply. If the min and the max of these are the same then case 2. If the min and max are different, case 3. Either way, the current pixel gets the min value of the neighbors. I skipped this step and applied it if case 1 doesn't apply, and update the EQ table iff case 3 is true.

```
1) Loop: i → [1 to numRows]
    a) Loop: j → [1 to numCols]
        i) pixel ← zeroFramedAry[i][j]
        ii) if pixel > 0
            (1) Array<> temp ← tempArr(set: 1, connectness: 4 or 8, i, j) // all neighbors
            (2) if case1(temp):
                (a) zeroFramedAry[i][j] ← ++newLabel
                (b) EQtable[newLabel] ← newLabel
            (3) Else
                (a) min ← minValue(temp)
                (b) zeroFramedAry[i][j] ← min
                (c) if case3(temp):
                    (i) Loop: t → temp:
                        1. EQtable[t] ← min
```

Array<> tempArr(set, connectness, i, j):
This function loads all the neighbors of the pixel into an array list, excluding zeros. Integer SET is used to switch the neighbors, either (b or g) and (d or e) for 4 -connectedness (Integer CONNECTNESS) and (a or f) and (c or h) for 8-connectedness.

Boolean case1(temp):
Case 1 checks if all the values in the array excluding the current pixel are zero. This is done by calculating the size of the array since any element in temp is not zero.

Boolean case3(temp):
Case 3 checks if the min and max from the neighbor array are not equal.

connectPass2(connectness):
Similar to pass1, it only used the connectness(4 or 8) to perform the algorithms second pass.

```
1) flag ← true
```

```
2) Loop: i → [numRows-1 → 1]
    a) Loop: j → [numCols-1 → 1]
        i) pixel ← zeroFramedAry[i][j]
        ii) if pixel > 0
            (1) Array<> temp ← tempArr(set: -1, connectness: 4 or 8, i, j) // all neighbors
            (2) Loop: t → temp:
                (a) If t != pixel :
                    (i) Flag ← false
            (3) If flag == false:
                (a) Min ← getMin(temp)
                (b) changeLabels(connectness, min, i, j)
                (c) zeroFramedAry[i][j] ← min
                (d) flag ← true
                (e) EQtable[pixel] ← min
```

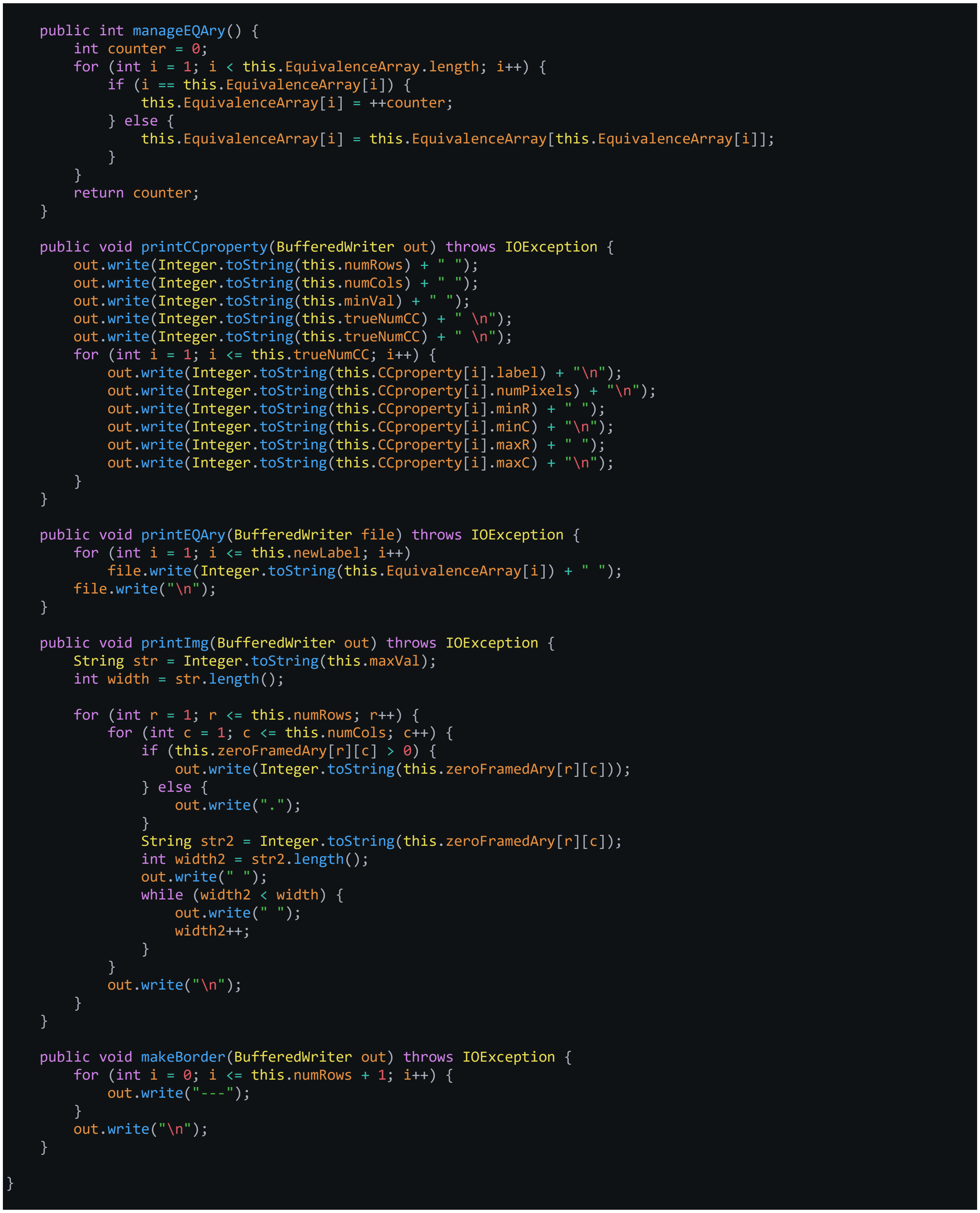
changeLabels(connectness, min, i, j):
This changes the neighbor elements if they are not equal to zero

```java
/*
Computer Vision
Project 5
Created by Adrian Noa

usage:

java adrian_noa_main.java 8 data1 pretty label property

*/

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

// import

public class CVProject5 {
    public static void main(String[] args) throws IOException {
        if (args.length != 5) {
            System.out.println("Not enough of arguments");
            return;
        }

        try (
            // Scanner inFile = new Scanner(new File(args[1]));
            Scanner inFile = readFile(args[1]);
            BufferedWriter prettyPrintFile = createFile(args[2]);
            BufferedWriter labelFile = createFile(args[3]);
            BufferedWriter propertyFile = createFile(args[4]);
        ) {

            ConnCompLabel connCompLabel = new ConnCompLabel(inFile);

            // step 1
            connCompLabel.zero2D(connCompLabel.zeroFramedAry);

            // step 2
            connCompLabel.loadImage(inFile);
```

```java
// step 3
int whichConnectness = Integer.parseInt(args[0]);

// step 4 & 5

connCompLabel.connectPass1(whichConnectness); // pass 1
prettyPrintFile.write("\n");
connCompLabel.makeBorder(prettyPrintFile);
prettyPrintFile.write("Result of Pass 1: \n");
connCompLabel.makeBorder(prettyPrintFile);
connCompLabel.imgReformat(prettyPrintFile);
connCompLabel.makeBorder(prettyPrintFile);
prettyPrintFile.write("Equivalence Array after Pass 1: \n");
connCompLabel.makeBorder(prettyPrintFile);
connCompLabel.printEQAry(prettyPrintFile);
connCompLabel.makeBorder(prettyPrintFile);
prettyPrintFile.write("\n");

connCompLabel.connectPass2(whichConnectness); // pass 2
prettyPrintFile.write("\n");
connCompLabel.makeBorder(prettyPrintFile);
prettyPrintFile.write("Result of Pass 2: \n");
connCompLabel.makeBorder(prettyPrintFile);
connCompLabel.imgReformat(prettyPrintFile);
connCompLabel.makeBorder(prettyPrintFile);
prettyPrintFile.write("Equivalence Array after Pass 2: \n");
connCompLabel.makeBorder(prettyPrintFile);
connCompLabel.printEQAry(prettyPrintFile);
connCompLabel.makeBorder(prettyPrintFile);
prettyPrintFile.write("\n");


// step 6
connCompLabel.trueNumCC = connCompLabel.manageEQAry();
connCompLabel.makeBorder(prettyPrintFile);
prettyPrintFile.write("Equivalence Array after management: \n");
connCompLabel.makeBorder(prettyPrintFile);
connCompLabel.printEQAry(prettyPrintFile);
connCompLabel.makeBorder(prettyPrintFile);
connCompLabel.newMin = 0;
connCompLabel.newMax = connCompLabel.trueNumCC;
connCompLabel.allocateCCproperty(connCompLabel.trueNumCC);

// step 7
connCompLabel.connectPass3(connCompLabel.zeroFramedAry,
connCompLabel.CCproperty);

// step 8
prettyPrintFile.write("\n");
connCompLabel.makeBorder(prettyPrintFile);
prettyPrintFile.write("Algorithm Pass 3 \n"); // pass 3
connCompLabel.makeBorder(prettyPrintFile);
connCompLabel.imgReformat(prettyPrintFile);

// step 9
connCompLabel.makeBorder(prettyPrintFile);
prettyPrintFile.write("Equivalence Array after Pass 3 \n");
connCompLabel.makeBorder(prettyPrintFile);
connCompLabel.printEQAry(prettyPrintFile);
connCompLabel.makeBorder(prettyPrintFile);

// step 10
labelFile.write(Integer.toString(connCompLabel.numRows)+" ");
labelFile.write(Integer.toString(connCompLabel.numCols)+" ");
labelFile.write(Integer.toString(connCompLabel.newMin)+" ");
labelFile.write(Integer.toString(connCompLabel.newMax)+"\n");

// step 11
connCompLabel.printImg(labelFile);

// step 12
connCompLabel.printCCproperty(propertyFile);
```

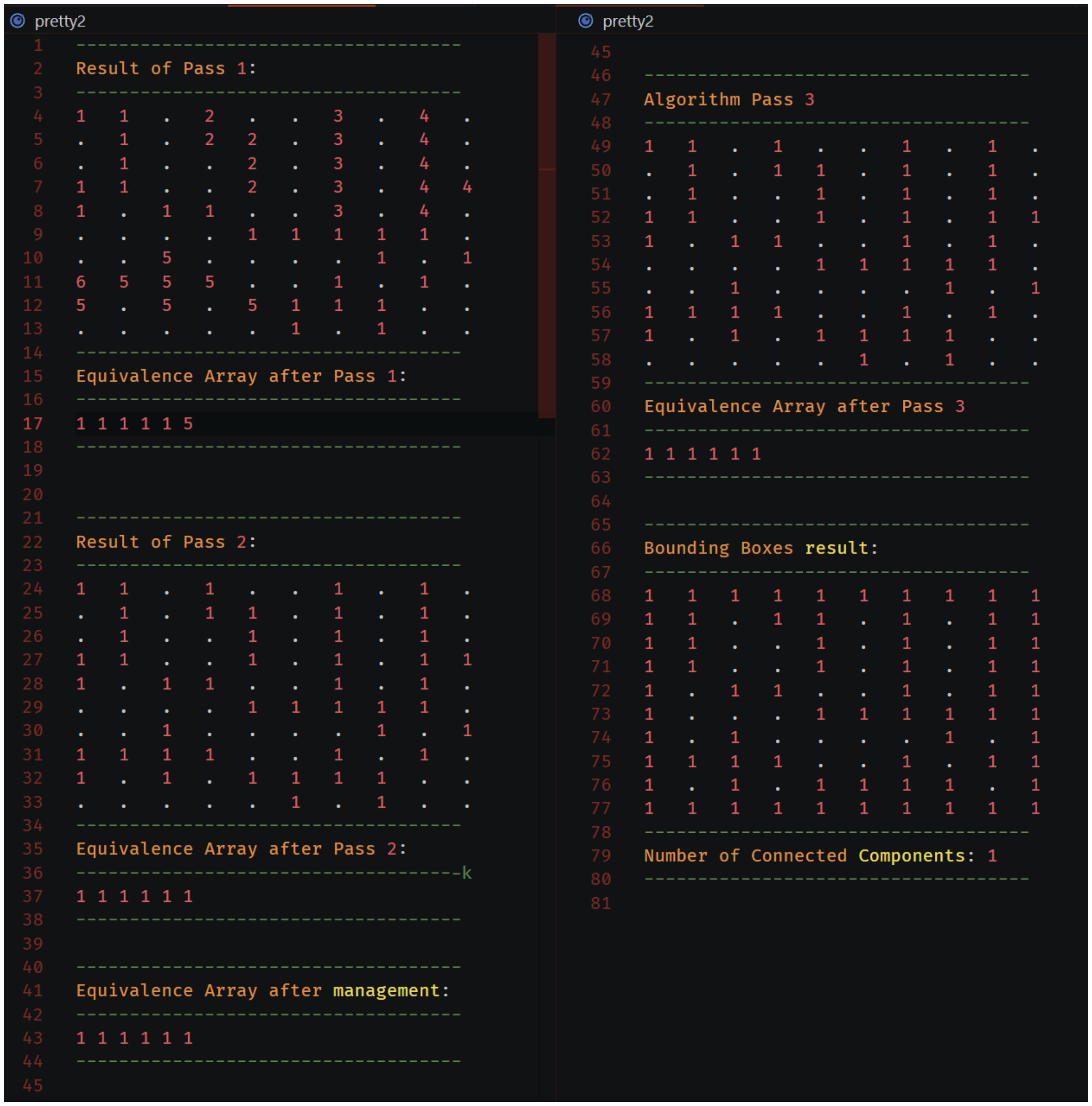
```java
            // step 13
            connCompLabel.drawBoxes(connCompLabel.CCproperty);

            // step 14
            prettyPrintFile.write("\n");
            connCompLabel.makeBorder(prettyPrintFile);
            prettyPrintFile.write("Bounding Boxes result:\n");
            connCompLabel.makeBorder(prettyPrintFile);
            connCompLabel.imgReformat(prettyPrintFile);
            connCompLabel.makeBorder(prettyPrintFile);

            // step 15
            prettyPrintFile.write("Number of Connected Components: " +
            Integer.toString(connCompLabel.trueNumCC)+"\n");
            connCompLabel.makeBorder(prettyPrintFile);

            System.out.println();

        }
    }
    public static Scanner readFile(String str) throws IOException{
        return new Scanner(new BufferedReader(new FileReader(str)));
    }

    public static BufferedWriter createFile(String str) throws IOException{
        return new BufferedWriter(new FileWriter(str));
    }
}

class Property {
    int label = 0;
    int numPixels = 0;
    int minR = 0;
    int minC = 0;
    int maxR = 0;
    int maxC = 0;
}

class ConnCompLabel extends Property {
    int numRows, numCols, minVal, maxVal, newLabel, newMin, newMax, trueNumCC;
    int zeroFramedAry[][];
    int nonZeroNeighborAry[];
    int EquivalenceArray[];
    Property[] CCproperty;

    ConnCompLabel(Scanner inFile) {
        this.numRows = inFile.nextInt();
        this.numCols = inFile.nextInt();
        this.minVal = inFile.nextInt();
        this.maxVal = inFile.nextInt();
        this.newLabel = 0;
        this.zeroFramedAry = new int[numRows + 2][numCols + 2];
        this.nonZeroNeighborAry = new int[5];
        this.EquivalenceArray = new int[(numRows * numCols) / 4];
    }

    public int[][] zero2D(int[][] arr) {
        /*
         * Initialize a 2-D array to zero
         */
        for (int i = 0; i < numRows; i++) {
            for (int j = 0; j < numCols; j++) {
                arr[i][j] = 0;
            }
        }
        return arr;
    }

    public void loadImage(Scanner b) throws IOException {
        /*
         * Read from input file and write to zeroFramedAry
         * begin at (1,1)
         */
        for (int i = 1; i <= numRows; i++) {
            for (int j = 1; j <= numCols; j++) {
                this.zeroFramedAry[i][j] = b.nextInt();
            }
        }
    }

    public void imgReformat(BufferedWriter file) throws IOException {
        for (int i = 1; i <= this.numRows; i++) {
            for (int j = 1; j <= this.numCols; j++) {
                if (this.zeroFramedAry[i][j] == 0)
                    file.write(".   ");
                else if(zeroFramedAry[i][j]>=10)
                    file.write(Integer.toString(this.zeroFramedAry[i][j])+"  ");
                else file.write(Integer.toString(this.zeroFramedAry[i][j])+"   ");
            }
            file.write("\n");
        }
    }


    public ArrayList<Integer> tempArr(int set, int connectness, int i, int j) {

        // a b c
        //
        // d P(i,j) e
        //
        // f g h
        ArrayList<Integer> temp = new ArrayList<Integer>();
        temp.add(this.zeroFramedAry[i + (-1)*set][j]); // b or g
        temp.add(this.zeroFramedAry[i][j + (-1)*set]); // d or e

        ArrayList<Integer> zero = new ArrayList<Integer>();
        zero.add(0);
        if (connectness == 8) {
            temp.add(this.zeroFramedAry[i + (-1)*set][j - 1]); // a or f
            temp.add(this.zeroFramedAry[i + (-1)*set][j + 1]); // c or h
        }
        temp.removeAll(zero);
        return temp;
    }

    public void changeLabels(int connectness, int val, int i, int j){
        if(this.zeroFramedAry[i + 1][j] != 0 )
            this.zeroFramedAry[i + 1][j] = val; // g

        if(this.zeroFramedAry[i][j + 1] != 0 )
            this.zeroFramedAry[i][j + 1] = val; // e

        if (connectness == 8) {
            if(this.zeroFramedAry[i + 1][j - 1] != 0 )
                this.zeroFramedAry[i + 1][j - 1] = val; // f
            if(this.zeroFramedAry[i + 1][j + 1] != 0 )
                this.zeroFramedAry[i + 1][j + 1] = val; // h
        }
    }

    public boolean case1(ArrayList<Integer> arr) {

        // if(Arrays.stream(arr).sum() == 0)
        if (arr.size() == 0) {
            System.out.print(" - case1 - all are zero\n");
            return true;
        }
        return false;
    }

    public boolean case2(ArrayList<Integer> arr) {

        // if( Arrays.stream(arr).min() == Arrays.stream(arr).max() ){
            if (arr.stream().mapToInt(Integer::intValue).min().getAsInt() ==
arr.stream().mapToInt(Integer::intValue).max()
```

```java
                .getAsInt()) {
                    System.out.print(" - case2\n");
                    return true;
        }
        return false;
    }

    public boolean case3(ArrayList<Integer> arr) {

        // if( Arrays.stream(arr).min() == Arrays.stream(arr).max() ){
            if (arr.stream().mapToInt(Integer::intValue).min().getAsInt() !=
arr.stream().mapToInt(Integer::intValue).max()
                .getAsInt()) {
                    System.out.print(" - case3\n");
                    return true;
        }
        return false;
    }

    public void printMap() {
        for (int i = 1; i <= this.numRows; i++) {
            for (int j = 1; j <= this.numCols; j++) {
                if (this.zeroFramedAry[i][j] == 0)
                    System.out.print(".  ");
                    else if(this.zeroFramedAry[i][j]>=10)
                    System.out.print(this.zeroFramedAry[i][j]+" ");
                    else System.out.print(this.zeroFramedAry[i][j]+"  ");
                }
                System.out.print("\n");
            }

    }


    public void connectPass1(int connectness) {
        for (int i = 1; i <= this.numRows; i++)
            for (int j = 1; j <= this.numCols; j++) {
                int pixel = this.zeroFramedAry[i][j];
                if (pixel > 0) {
                    ArrayList<Integer> temp = tempArr(1, connectness, i, j);
                        if (case1(temp)) {
                            this.zeroFramedAry[i][j] = ++this.newLabel;
                            updateEquivalencyTable(this.newLabel, this.newLabel);
                        } else {
                            int min = temp.stream().mapToInt(Integer::intValue).min().getAsInt();

                            this.zeroFramedAry[i][j] = min;

                            if(case3(temp)){
                                for(int t:temp)
                                    updateEquivalencyTable(t,min);
                            }
                    }
                }
            }
    }

    public void connectPass2(int connectness) {

        boolean flag = true;
        for (int i = this.numRows-1; i >= 1; i--) {
            for (int j = this.numCols-1; j >= 1; j--) {
                int pixel = zeroFramedAry[i][j];
                if (pixel > 0) {
                    ArrayList<Integer> temp = tempArr(-1, connectness, i, j);
                    for(int t:temp) if(t!=pixel) flag=false;
                    // case 3
                    if(!flag) {
                        int min = temp.stream().mapToInt(Integer::intValue).min().getAsInt();
                        changeLabels(connectness, min, i, j);
                        zeroFramedAry[i][j] = min;
                        flag = true;
                        updateEquivalencyTable(pixel, min);
                    }
                }
            }
        }
    }

    public void connectPass3(int[][] inArr, Property[] CCproperty) {
        int k = 0;
        int nummPixs;
        int pixel;
        for (int i = 1; i <= this.trueNumCC; i++) {
            this.CCproperty[i].label = i;
            this.CCproperty[i].numPixels = 0;
            this.CCproperty[i].minR = this.numRows;
            this.CCproperty[i].maxR = 0;
            this.CCproperty[i].minC = this.numCols;
            this.CCproperty[i].maxC = 0;
        }
        for (int r = 1; r <= this.numRows; r++) {
            for (int c = 1; c <= this.numCols; c++) {
                pixel = inArr[r][c];
                if (pixel > 0) {
                    inArr[r][c] = this.EquivalenceArray[pixel];
                    k = inArr[r][c];
                    nummPixs = this.CCproperty[k].numPixels;
                    CCproperty[k].numPixels = ++nummPixs;
                    if (r < this.CCproperty[k].minR) {
                        this.CCproperty[k].minR = r - 1;
                    }
                    if (r > this.CCproperty[k].maxR) {
                        this.CCproperty[k].maxR = r - 1;
                    }
                    if (c < this.CCproperty[k].minC) {
                        this.CCproperty[k].minC = c - 1;
                    }
                    if (c > this.CCproperty[k].maxC) {
                        this.CCproperty[k].maxC = c - 1;
                    }
                }
            }
        }
    }

    public void allocateCCproperty(int truenumcc) {
        this.CCproperty = new Property[truenumcc + 1];
        for (int i = 0; i <= truenumcc; i++) {
            this.CCproperty[i] = new Property();
        }
    }

    public void drawBoxes(Property[] CCproperty) {
        int minRow, minCol, maxRow, maxCol, label;
        for (int index = 1; index <= this.trueNumCC; index++) {
            minRow = this.CCproperty[index].minR + 1;
            minCol = this.CCproperty[index].minC + 1;
            maxRow = this.CCproperty[index].maxR + 1;
            maxCol = this.CCproperty[index].maxC + 1;
            label = this.CCproperty[index].label;

            for (int i = minCol; i <= maxCol; i++) {
                this.zeroFramedAry[minRow][i] = label;
                this.zeroFramedAry[maxRow][i] = label;
            }
            for (int i = minRow; i <= maxRow; i++) {
                this.zeroFramedAry[i][minCol] = label;
                this.zeroFramedAry[i][maxCol] = label;
            }
        }
    }

    public void updateEquivalencyTable(int index, int label) {
        this.EquivalenceArray[index] = label;
    }
```



```java
    public int manageEQAry() {
        int counter = 0;
        for (int i = 1; i < this.EquivalenceArray.length; i++) {
            if (i == this.EquivalenceArray[i]) {
                this.EquivalenceArray[i] = ++counter;
            } else {
                this.EquivalenceArray[i] = this.EquivalenceArray[this.EquivalenceArray[i]];
            }
        }
        return counter;
    }

    public void printCCproperty(BufferedWriter out) throws IOException {
        out.write(Integer.toString(this.numRows) + " ");
        out.write(Integer.toString(this.numCols) + " ");
        out.write(Integer.toString(this.minVal) + " ");
        out.write(Integer.toString(this.trueNumCC) + " \n");
        out.write(Integer.toString(this.trueNumCC) + " \n");
        for (int i = 1; i <= this.trueNumCC; i++) {
            out.write(Integer.toString(this.CCproperty[i].label) + "\n");
            out.write(Integer.toString(this.CCproperty[i].numPixels) + "\n");
            out.write(Integer.toString(this.CCproperty[i].minR) + " ");
            out.write(Integer.toString(this.CCproperty[i].minC) + "\n");
            out.write(Integer.toString(this.CCproperty[i].maxR) + " ");
            out.write(Integer.toString(this.CCproperty[i].maxC) + "\n");
        }
    }

    public void printEQAry(BufferedWriter file) throws IOException {
        for (int i = 1; i <= this.newLabel; i++)
            file.write(Integer.toString(this.EquivalenceArray[i]) + " ");
        file.write("\n");
    }

    public void printImg(BufferedWriter out) throws IOException {
        String str = Integer.toString(this.maxVal);
        int width = str.length();

        for (int r = 1; r <= this.numRows; r++) {
            for (int c = 1; c <= this.numCols; c++) {
                if (this.zeroFramedAry[r][c] > 0) {
                    out.write(Integer.toString(this.zeroFramedAry[r][c]));
                } else {
                    out.write(".");
                }
                String str2 = Integer.toString(this.zeroFramedAry[r][c]);
                int width2 = str2.length();
                out.write(" ");
                while (width2 < width) {
                    out.write(" ");
                    width2++;
                }
            }
            out.write("\n");
        }
    }

    public void makeBorder(BufferedWriter out) throws IOException {
        for (int i = 0; i <= this.numRows + 1; i++) {
            out.write("---");
        }
        out.write("\n");
    }

}
```

Data1 8-connectness outputs

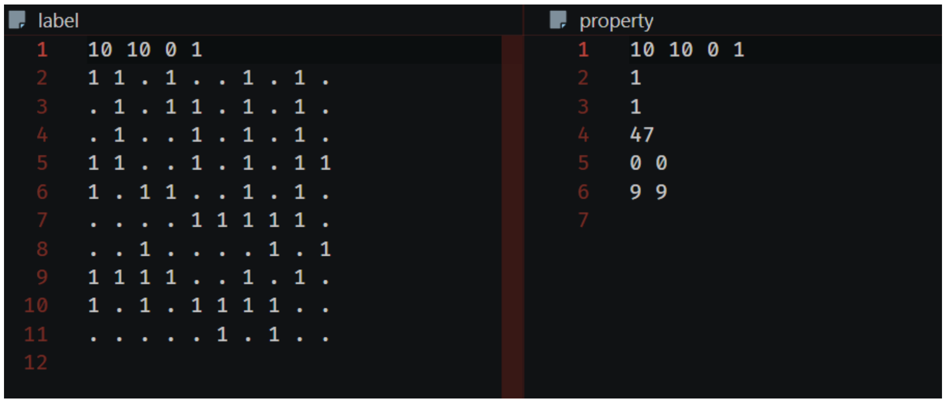


pretty2

```
------------------------------------
Result of Pass 1:
------------------------------------
1   1   .   2   .   .   3   .   4   .
.   1   .   2   2   .   3   .   4   .
.   1   .   .   2   .   3   .   4   .
1   1   .   .   2   .   3   .   4   4
1   .   1   1   .   .   3   .   4   .
.   .   .   .   1   1   1   1   1   .
.   .   5   .   .   .   .   1   .   1
6   5   5   5   .   .   1   .   1   .
5   .   5   .   5   1   1   1   .   .
.   .   .   .   .   1   .   1   .   .
------------------------------------
Equivalence Array after Pass 1:
------------------------------------
1 1 1 1 1 5
------------------------------------


------------------------------------
Result of Pass 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   .   .   .   .   1   .   1
1   1   1   1   .   .   1   .   1   .
1   .   1   .   1   1   1   1   .   .
.   .   .   .   .   1   .   1   .   .
------------------------------------
Equivalence Array after Pass 2:
------------------------------------k
1 1 1 1 1 1
------------------------------------

------------------------------------
Equivalence Array after management:
------------------------------------
1 1 1 1 1 1
------------------------------------
```

pretty2

```

------------------------------------
Algorithm Pass 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   1   1   1   1   .
.   .   1   .   .   .   .   1   .   1
1   1   1   1   .   .   1   .   1   .
1   .   1   .   1   1   1   1   .   .
.   .   .   .   .   1   .   1   .   .
------------------------------------
Equivalence Array after Pass 3
------------------------------------
1 1 1 1 1 1
------------------------------------

------------------------------------
Bounding Boxes result:
------------------------------------
1   1   1   1   1   1   1   1   1   1
1   1   .   1   1   .   1   .   1   1
1   1   .   .   1   .   1   .   1   1
1   1   .   .   1   .   1   .   1   1
1   .   1   1   .   .   1   .   1   1
1   .   .   .   1   1   1   1   1   1
1   .   1   .   .   .   .   1   .   1
1   1   1   1   .   .   1   .   1   1
1   .   1   .   1   1   1   1   .   1
1   1   1   1   1   1   1   1   1   1
------------------------------------
Number of Connected Components: 1
------------------------------------
```



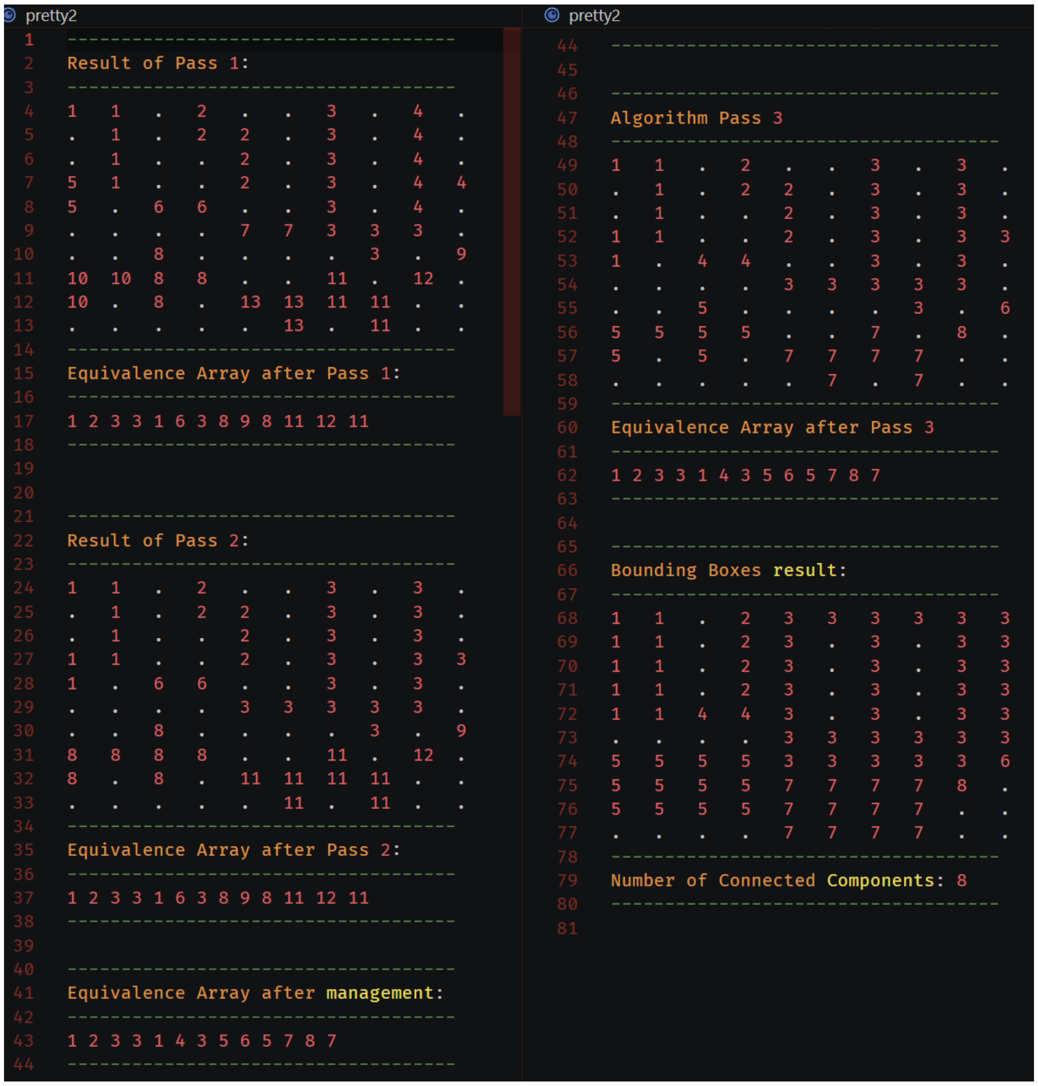
label

```
10 10 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 . .
```

property

```
10 10 0 1
1
1
47
0 0
9 9
```

Data1 4-Connectedness outputs

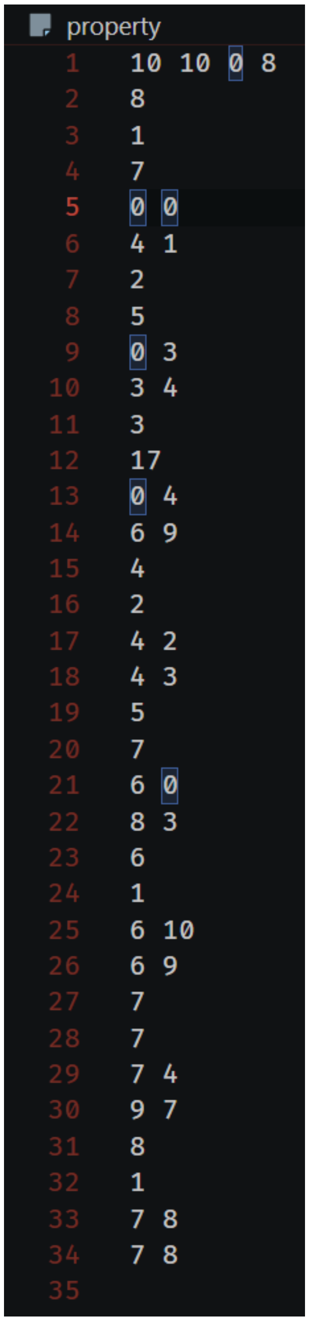


pretty2

```
------------------------------------
Result of Pass 1:
------------------------------------
1   1   .   2   .   .   3   .   4   .
.   1   .   2   2   .   3   .   4   .
.   1   .   .   2   .   3   .   4   .
5   1   .   .   2   .   3   .   4   4
5   .   6   6   .   .   3   .   4   .
.   .   .   .   7   7   3   3   3   .
.   .   8   .   .   .   .   3   .   9
10  10  8   8   .   .   11  .   12  .
10  .   8   .   13  13  11  11  .   .
.   .   .   .   .   13  .   11  .   .
------------------------------------
Equivalence Array after Pass 1:
------------------------------------
1 2 3 3 1 6 3 8 9 8 11 12 11
------------------------------------


------------------------------------
Result of Pass 2:
------------------------------------
1   1   .   2   .   .   3   .   3   .
.   1   .   2   2   .   3   .   3   .
.   1   .   .   2   .   3   .   3   .
1   1   .   .   2   .   3   .   3   3
1   .   6   6   .   .   3   .   3   .
.   .   .   .   3   3   3   3   3   .
.   .   8   .   .   .   .   3   .   9
8   8   8   8   .   .   11  .   12  .
8   .   8   .   11  11  11  11  .   .
.   .   .   .   .   11  .   11  .   .
------------------------------------
Equivalence Array after Pass 2:
------------------------------------
1 2 3 3 1 6 3 8 9 8 11 12 11
------------------------------------


------------------------------------
Equivalence Array after management:
------------------------------------
1 2 3 3 1 4 3 5 6 5 7 8 7
------------------------------------
```

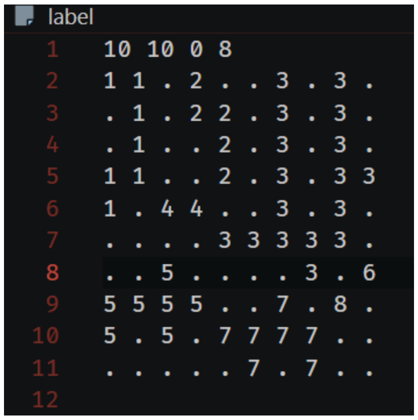
pretty2

```
------------------------------------

------------------------------------
Algorithm Pass 3
------------------------------------
1   1   .   2   .   .   3   .   3   .
.   1   .   2   2   .   3   .   3   .
.   1   .   .   2   .   3   .   3   .
1   1   .   .   2   .   3   .   3   3
1   .   4   4   .   .   3   .   3   .
.   .   .   .   3   3   3   3   3   .
.   .   5   .   .   .   .   3   .   6
5   5   5   5   .   .   7   .   8   .
5   .   5   .   7   7   7   7   .   .
.   .   .   .   .   7   .   7   .   .
------------------------------------
Equivalence Array after Pass 3
------------------------------------
1 2 3 3 1 4 3 5 6 5 7 8 7
------------------------------------


------------------------------------
Bounding Boxes result:
------------------------------------
1   1   .   2   3   3   3   3   3   3
1   1   .   2   3   .   3   .   3   3
1   1   .   2   3   .   3   .   3   3
1   1   .   2   3   .   3   .   3   3
1   1   4   4   3   .   3   .   3   3
.   .   .   .   3   3   3   3   3   3
5   5   5   5   3   3   3   3   3   6
5   5   5   5   7   7   7   7   8   .
5   5   5   5   7   7   7   7   .   .
.   .   .   .   7   7   7   7   .   .
------------------------------------
Number of Connected Components: 8
------------------------------------
```



property

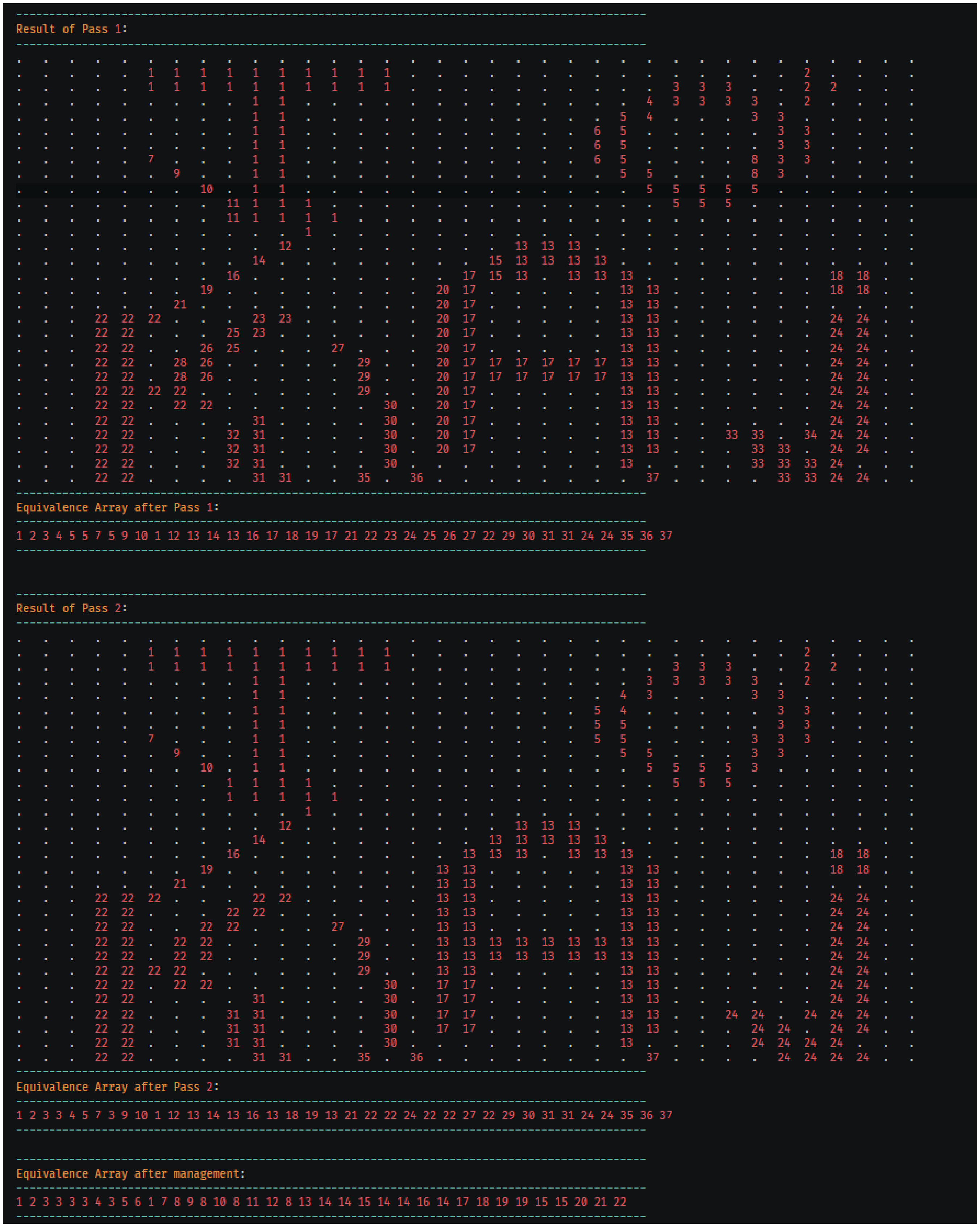
```
10 10 0 8
8
1
7
0 0
4 1
2
5
0 3
3 4
3
17
0 4
6 9
4
2
4 2
4 3
5
7
6 0
8 3
6
1
6 10
6 9
7
7
7 4
9 7
8
1
7 8
7 8
```



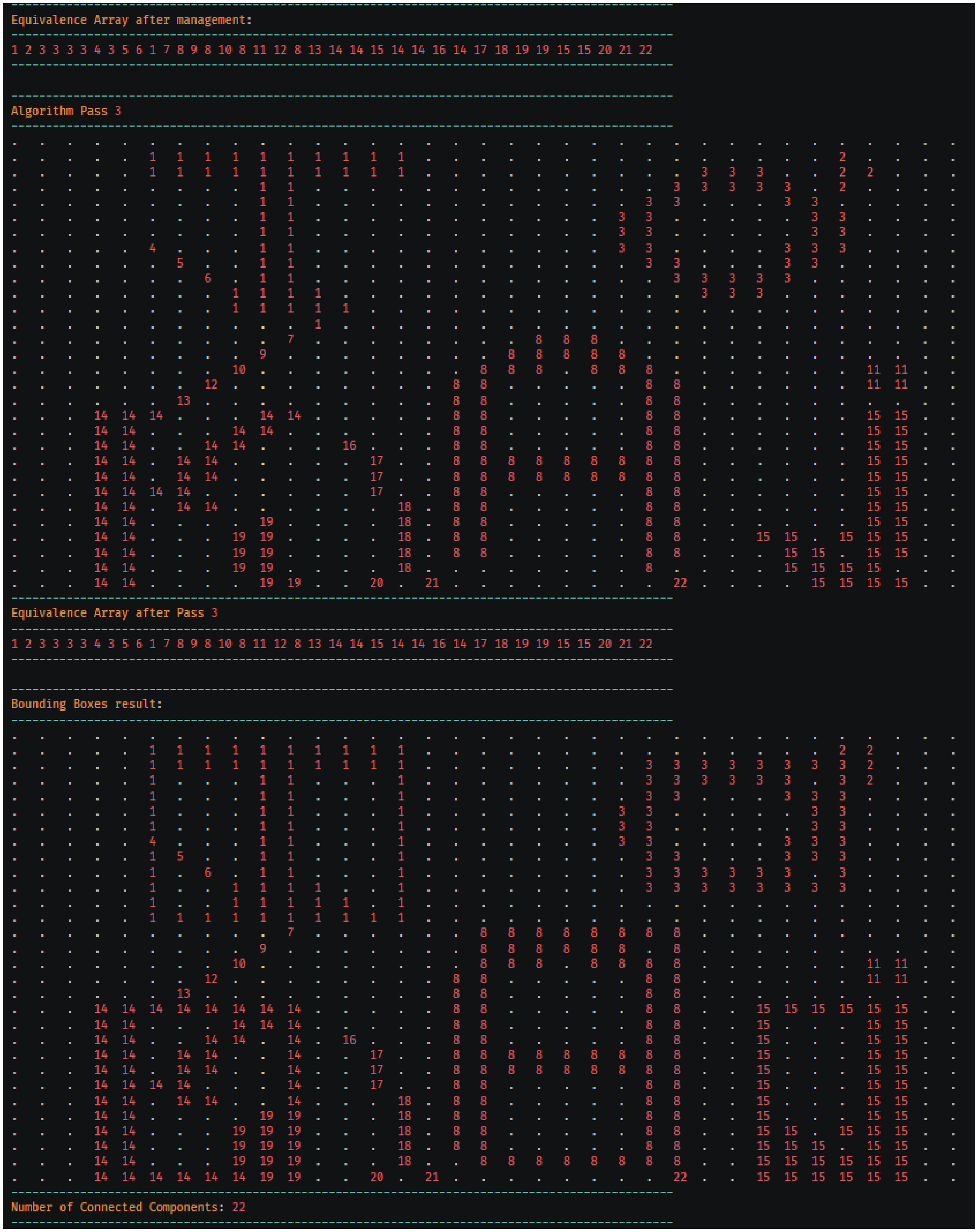
label

```
10 10 0 8
1 1 . 2 . . 3 . 3 .
. 1 . 2 2 . 3 . 3 .
. 1 . . 2 . 3 . 3 .
1 1 . . 2 . 3 . 3 3
1 . 4 4 . . 3 . 3 .
. . . . 3 3 3 3 3 .
. . 5 . . . . 3 . 6
5 5 5 5 . . 7 . 8 .
5 . 5 . 7 7 7 7 . .
. . . . . 7 . 7 . .
```

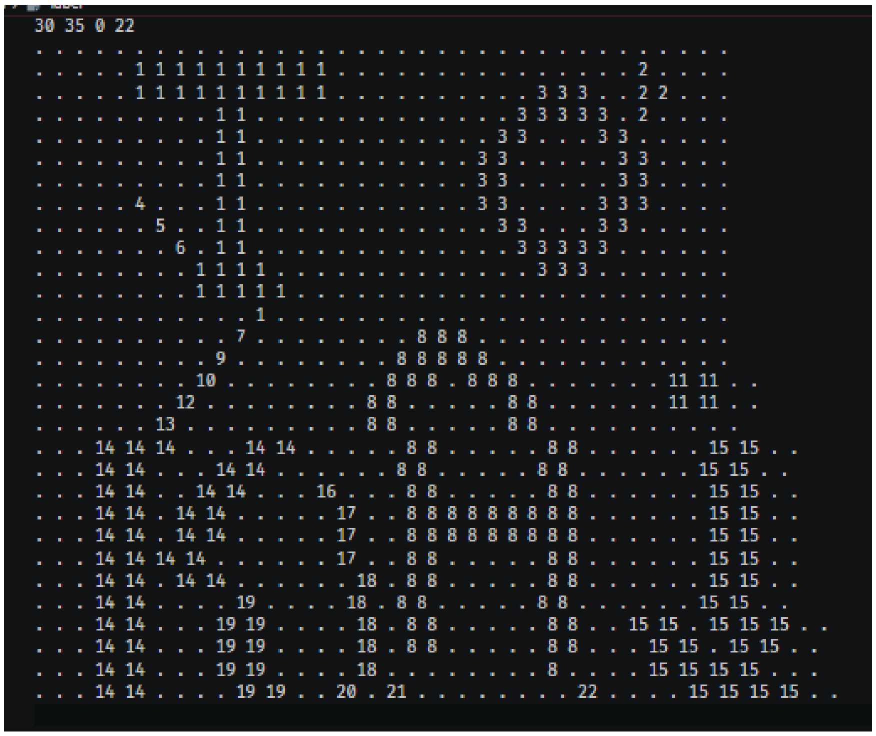
Data2 4-connectness outputs:



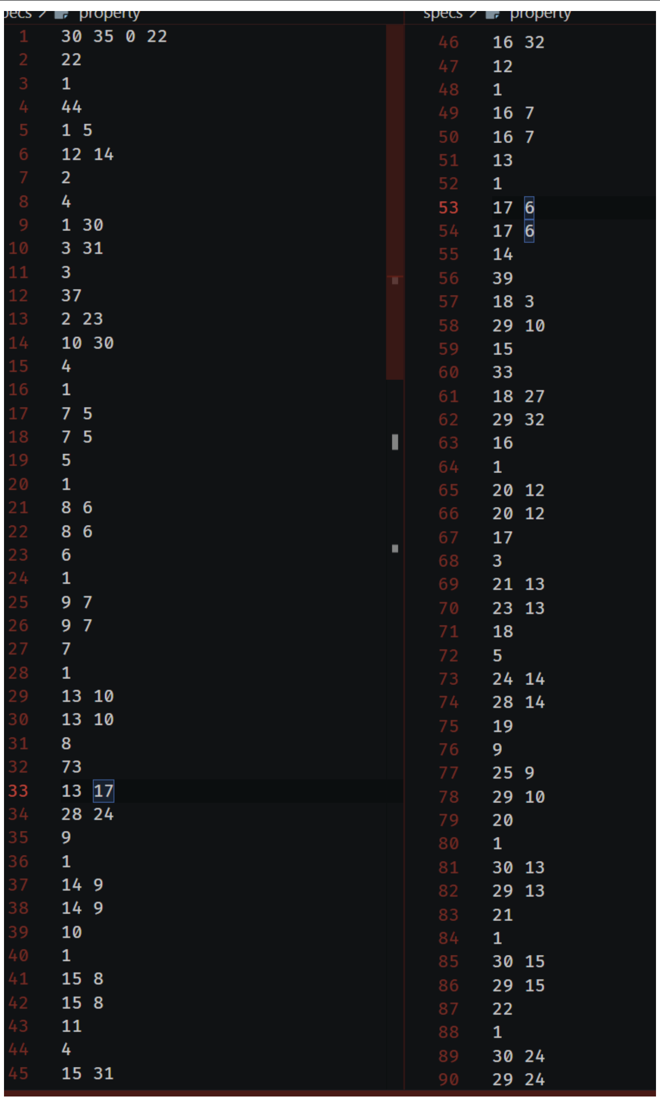
```
------------------------------------------------------------------------------------------------
Result of Pass 1:
------------------------------------------------------------------------------------------------
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   2   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   3   3   3   .   .   2   2   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   4   3   3   3   3   .   2   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   5   4   .   .   .   3   3   .   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   6   5   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   6   5   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   7   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   6   5   .   .   .   .   8   3   3   .   .   .   .
.   .   .   .   .   .   9   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   5   5   .   .   .   8   3   .   .   .   .   .
.   .   .   .   .   .   .   10  .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   5   5   5   5   5   .   .   .   .   .   .
.   .   .   .   .   .   .   .   11  1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   5   5   5   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   11  1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   12  .   .   .   .   .   .   .   .   13  13  13  .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   14  .   .   .   .   .   .   .   .   15  13  13  13  13  .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   16  .   .   .   .   .   .   .   .   17  15  13  .   13  13  13  .   .   .   .   .   .   .   18  18  .   .
.   .   .   .   .   .   .   19  .   .   .   .   .   .   .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   18  18  .   .
.   .   .   .   .   .   21  .   .   .   .   .   .   .   .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   .   .   .   .
.   .   .   22  22  22  .   .   .   23  23  .   .   .   .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   25  23  .   .   .   .   .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   26  25  .   .   .   27  .   .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   28  26  .   .   .   .   .   29  .   .   20  17  17  17  17  17  17  13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   28  26  .   .   .   .   .   29  .   .   20  17  17  17  17  17  17  13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  22  22  .   .   .   .   .   .   29  .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   22  22  .   .   .   .   .   .   30  .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   .   31  .   .   .   .   30  .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   32  31  .   .   .   .   30  .   20  17  .   .   .   .   .   13  13  .   .   33  33  .   34  24  24  .   .
.   .   .   22  22  .   .   .   32  31  .   .   .   .   30  .   20  17  .   .   .   .   .   13  13  .   .   .   33  33  .   24  24  .   .
.   .   .   22  22  .   .   .   32  31  .   .   .   .   30  .   .   .   .   .   .   .   .   13  .   .   .   .   33  33  33  24  .   .   .
.   .   .   22  22  .   .   .   .   31  31  .   .   35  .   36  .   .   .   .   .   .   .   .   37  .   .   .   .   33  33  24  24  .   .
------------------------------------------------------------------------------------------------
Equivalence Array after Pass 1:
------------------------------------------------------------------------------------------------
1 2 3 4 5 5 7 5 9 10 1 12 13 14 13 16 17 18 19 17 21 22 23 24 25 26 27 22 29 30 31 31 24 24 35 36 37
------------------------------------------------------------------------------------------------


------------------------------------------------------------------------------------------------
Result of Pass 2:
------------------------------------------------------------------------------------------------
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   2   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   3   3   3   .   .   2   2   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   3   3   3   3   3   .   2   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   4   3   .   .   .   3   3   .   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   5   4   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   5   5   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   7   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   5   5   .   .   .   .   3   3   3   .   .   .   .
.   .   .   .   .   .   9   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   5   5   .   .   .   3   3   .   .   .   .   .
.   .   .   .   .   .   .   10  .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   5   5   5   5   3   .   .   .   .   .   .
.   .   .   .   .   .   .   .   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   5   5   5   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   12  .   .   .   .   .   .   .   .   13  13  13  .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   14  .   .   .   .   .   .   .   .   13  13  13  13  13  .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   16  .   .   .   .   .   .   .   .   13  13  13  .   13  13  13  .   .   .   .   .   .   .   18  18  .   .
.   .   .   .   .   .   .   19  .   .   .   .   .   .   .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   18  18  .   .
.   .   .   .   .   .   21  .   .   .   .   .   .   .   .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   .   .   .   .
.   .   .   22  22  22  .   .   .   22  22  .   .   .   .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   22  22  .   .   .   .   .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   22  22  .   .   .   27  .   .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   22  22  .   .   .   .   .   29  .   .   13  13  13  13  13  13  13  13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   22  22  .   .   .   .   .   29  .   .   13  13  13  13  13  13  13  13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  22  22  .   .   .   .   .   .   29  .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   22  22  .   .   .   .   .   .   30  .   17  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   .   31  .   .   .   .   30  .   17  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   31  31  .   .   .   .   30  .   17  17  .   .   .   .   .   13  13  .   .   24  24  .   24  24  24  .   .
.   .   .   22  22  .   .   .   31  31  .   .   .   .   30  .   17  17  .   .   .   .   .   13  13  .   .   .   24  24  .   24  24  .   .
.   .   .   22  22  .   .   .   31  31  .   .   .   .   30  .   .   .   .   .   .   .   .   13  .   .   .   .   24  24  24  24  .   .   .
.   .   .   22  22  .   .   .   .   31  31  .   .   35  .   36  .   .   .   .   .   .   .   .   37  .   .   .   .   24  24  24  24  .   .
------------------------------------------------------------------------------------------------
Equivalence Array after Pass 2:
------------------------------------------------------------------------------------------------
1 2 3 3 4 5 7 3 9 10 1 12 13 14 13 16 13 18 19 13 21 22 22 24 22 22 27 22 29 30 31 31 24 24 35 36 37
------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------
Equivalence Array after management:
------------------------------------------------------------------------------------------------
1 2 3 3 3 3 4 3 5 6 1 7 8 9 8 10 8 11 12 8 13 14 14 15 14 14 16 14 17 18 19 19 15 15 20 21 22
------------------------------------------------------------------------------------------------
```



```
----------------------------------------------------------------------------------------------
Equivalence Array after management:
----------------------------------------------------------------------------------------------
1 2 3 3 3 3 4 3 5 6 1 7 8 9 8 10 8 11 12 8 13 14 14 15 14 14 16 14 17 18 19 19 15 15 20 21 22
----------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------
Algorithm Pass 3
----------------------------------------------------------------------------------------------
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   2   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   3   3   3   .   .   2   2   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   3   3   3   3   3   .   2   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   3   3   .   .   .   3   3   .   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   3   3   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   3   3   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   4   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   3   3   .   .   .   .   3   3   3   .   .   .   .
.   .   .   .   .   .   5   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   3   3   .   .   .   3   3   .   .   .   .   .
.   .   .   .   .   .   .   6   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   3   3   3   3   3   .   .   .   .   .   .
.   .   .   .   .   .   .   .   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   3   3   3   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   7   .   .   .   .   .   .   .   .   8   8   8   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   9   .   .   .   .   .   .   .   .   8   8   8   8   8   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   10  .   .   .   .   .   .   .   .   8   8   8   .   8   8   8   .   .   .   .   .   .   .   11  11  .   .
.   .   .   .   .   .   .   12  .   .   .   .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   11  11  .   .
.   .   .   .   .   .   13  .   .   .   .   .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   .   .   .   .
.   .   .   14  14  14  .   .   .   14  14  .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   .   .   14  14  .   .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   .   14  14  .   .   .   16  .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   .   .   .   17  .   .   8   8   8   8   8   8   8   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   .   .   .   17  .   .   8   8   8   8   8   8   8   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  14  14  .   .   .   .   .   .   17  .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   .   .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   .   .   .   19  .   .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   .   .   19  19  .   .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   15  15  .   15  15  15  .   .
.   .   .   14  14  .   .   .   19  19  .   .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   .   15  15  .   15  15  .   .
.   .   .   14  14  .   .   .   19  19  .   .   .   .   18  .   .   .   .   .   .   .   .   8   .   .   .   .   15  15  15  15  .   .   .
.   .   .   14  14  .   .   .   .   19  19  .   .   20  .   21  .   .   .   .   .   .   .   .   22  .   .   .   .   15  15  15  15  .   .
----------------------------------------------------------------------------------------------
Equivalence Array after Pass 3
----------------------------------------------------------------------------------------------
1 2 3 3 3 3 4 3 5 6 1 7 8 9 8 10 8 11 12 8 13 14 14 15 14 14 16 14 17 18 19 19 15 15 20 21 22
----------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------
Bounding Boxes result:
----------------------------------------------------------------------------------------------
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   2   2   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   3   3   3   3   3   3   3   3   2   .   .   .
.   .   .   .   .   1   .   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   .   3   3   3   3   3   3   .   3   2   .   .   .
.   .   .   .   .   1   .   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   .   3   3   .   .   .   3   3   3   .   .   .   .
.   .   .   .   .   1   .   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   3   3   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   1   .   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   3   3   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   4   .   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   3   3   .   .   .   .   3   3   3   .   .   .   .
.   .   .   .   .   1   5   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   .   3   3   .   .   .   3   3   3   .   .   .   .
.   .   .   .   .   1   .   6   .   1   1   .   .   .   1   .   .   .   .   .   .   .   .   3   3   3   3   3   3   .   3   .   .   .   .
.   .   .   .   .   1   .   .   1   1   1   1   .   .   1   .   .   .   .   .   .   .   .   3   3   3   3   3   3   3   3   .   .   .   .
.   .   .   .   .   1   .   .   1   1   1   1   1   .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   7   .   .   .   .   .   .   8   8   8   8   8   8   8   8   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   9   .   .   .   .   .   .   .   8   8   8   8   8   8   .   8   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   10  .   .   .   .   .   .   .   .   8   8   8   .   8   8   8   8   .   .   .   .   .   .   11  11  .   .
.   .   .   .   .   .   .   12  .   .   .   .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   11  11  .   .
.   .   .   .   .   .   13  .   .   .   .   .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   .   .   .   .
.   .   .   14  14  14  14  14  14  14  14  .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   15  15  15  15  15  15  .   .
.   .   .   14  14  .   .   .   14  14  14  .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   .   14  14  .   14  .   16  .   .   .   8   8   .   .   .   .   .   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   14  .   .   17  .   .   8   8   8   8   8   8   8   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   14  .   .   17  .   .   8   8   8   8   8   8   8   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  14  14  .   .   .   14  .   .   17  .   .   8   8   .   .   .   .   .   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   14  .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   .   .   .   19  19  .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   .   .   19  19  19  .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   15  15  .   15  15  15  .   .
.   .   .   14  14  .   .   .   19  19  19  .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   15  15  15  .   15  15  .   .
.   .   .   14  14  .   .   .   19  19  19  .   .   .   18  .   .   8   8   8   8   8   8   8   8   .   .   15  15  15  15  15  15  .   .
.   .   .   14  14  14  14  14  14  19  19  .   .   20  .   21  .   .   .   .   .   .   .   .   22  .   .   15  15  15  15  15  15  .   .
----------------------------------------------------------------------------------------------
Number of Connected Components: 22
----------------------------------------------------------------------------------------------
```

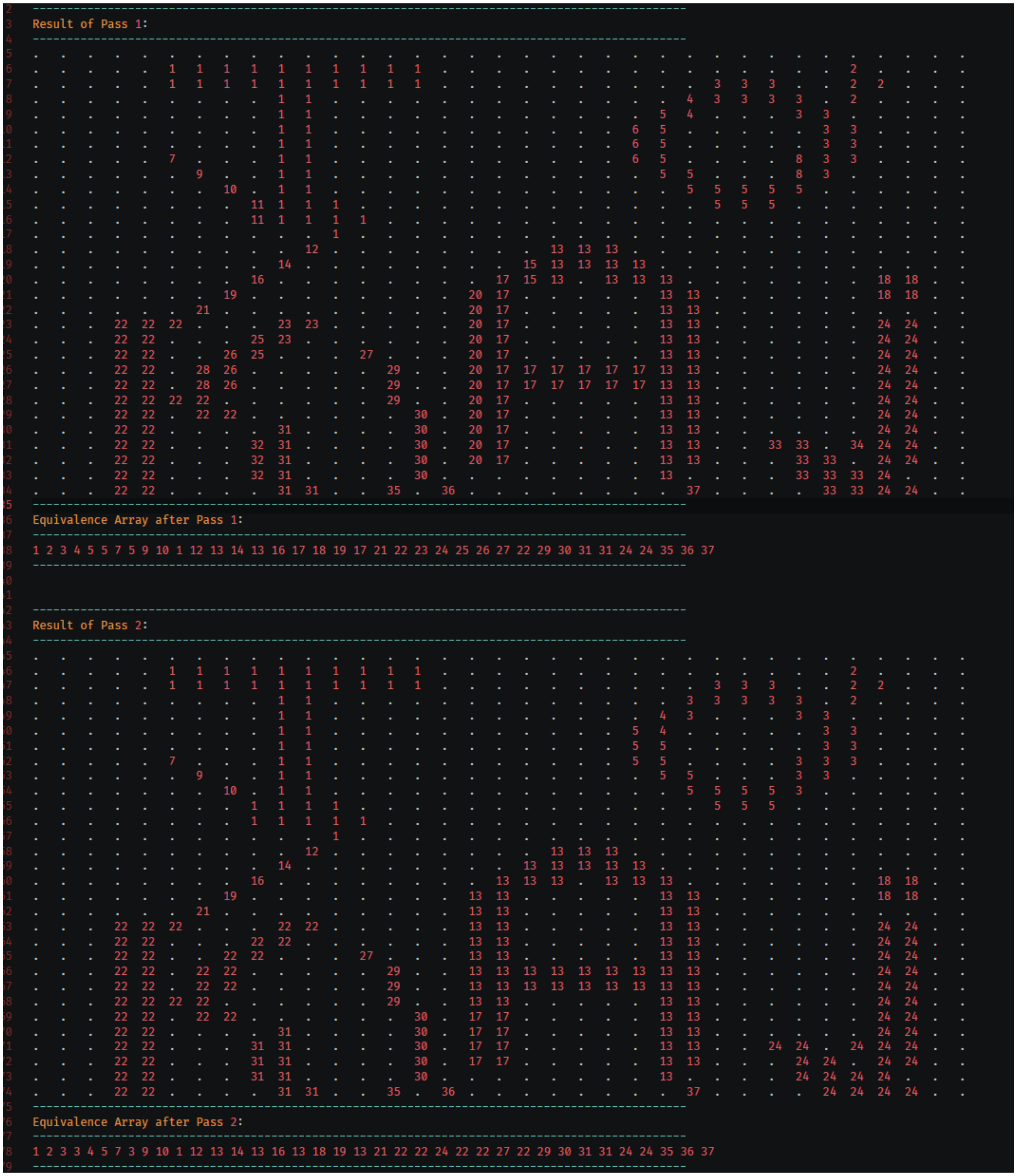


```
30 35 0 22
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 . . . . . . . . . . . . . . . 2 . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 . . . . . . . . . . 3 3 3 . . 2 2 . . .
. . . . . . . . . 1 1 . . . . . . . . . . . . . 3 3 3 3 3 . 2 . . . .
. . . . . . . . . 1 1 . . . . . . . . . . . . 3 3 . . . 3 3 . . . . .
. . . . . . . . . 1 1 . . . . . . . . . . . 3 3 . . . . . 3 3 . . . .
. . . . . . . . . 1 1 . . . . . . . . . . . 3 3 . . . . . 3 3 . . . .
. . . . . 4 . . . 1 1 . . . . . . . . . . . 3 3 . . . . 3 3 3 . . . .
. . . . . . 5 . . 1 1 . . . . . . . . . . . . 3 3 . . . 3 3 . . . . .
. . . . . . . 6 . 1 1 . . . . . . . . . . . . . 3 3 3 3 3 . . . . . .
. . . . . . . . 1 1 1 1 . . . . . . . . . . . . . 3 3 3 . . . . . . .
. . . . . . . . 1 1 1 1 1 . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . 7 . . . . . . . . 8 8 8 . . . . . . . . . . . . .
. . . . . . . . . 9 . . . . . . . . 8 8 8 8 8 . . . . . . . . . . . .
. . . . . . . . 10 . . . . . . . . 8 8 8 . 8 8 8 . . . . . . . 11 11 . .
. . . . . . . 12 . . . . . . . . 8 8 . . . . . 8 8 . . . . . . 11 11 . .
. . . . . . 13 . . . . . . . . . 8 8 . . . . . 8 8 . . . . . . . . . .
. . . 14 14 14 . . . 14 14 . . . . . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . . . 14 14 . . . . . . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . . 14 14 . . . 16 . . . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . 14 14 . . . . . 17 . . 8 8 8 8 8 8 8 8 8 . . . . . . 15 15 . .
. . . 14 14 . 14 14 . . . . . 17 . . 8 8 8 8 8 8 8 8 8 . . . . . . 15 15 . .
. . . 14 14 14 14 . . . . . . 17 . . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . 14 14 . . . . . . 18 . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . . . . 19 . . . . 18 . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . . . 19 19 . . . . 18 . 8 8 . . . . . 8 8 . . 15 15 . 15 15 15 . .
. . . 14 14 . . . 19 19 . . . . 18 . 8 8 . . . . . 8 8 . . . 15 15 . 15 15 . .
. . . 14 14 . . . 19 19 . . . . 18 . . . . . . . . 8 . . . . 15 15 15 15 . . .
. . . 14 14 . . . . 19 19 . . 20 . 21 . . . . . . . . 22 . . . . 15 15 15 15 . .
```

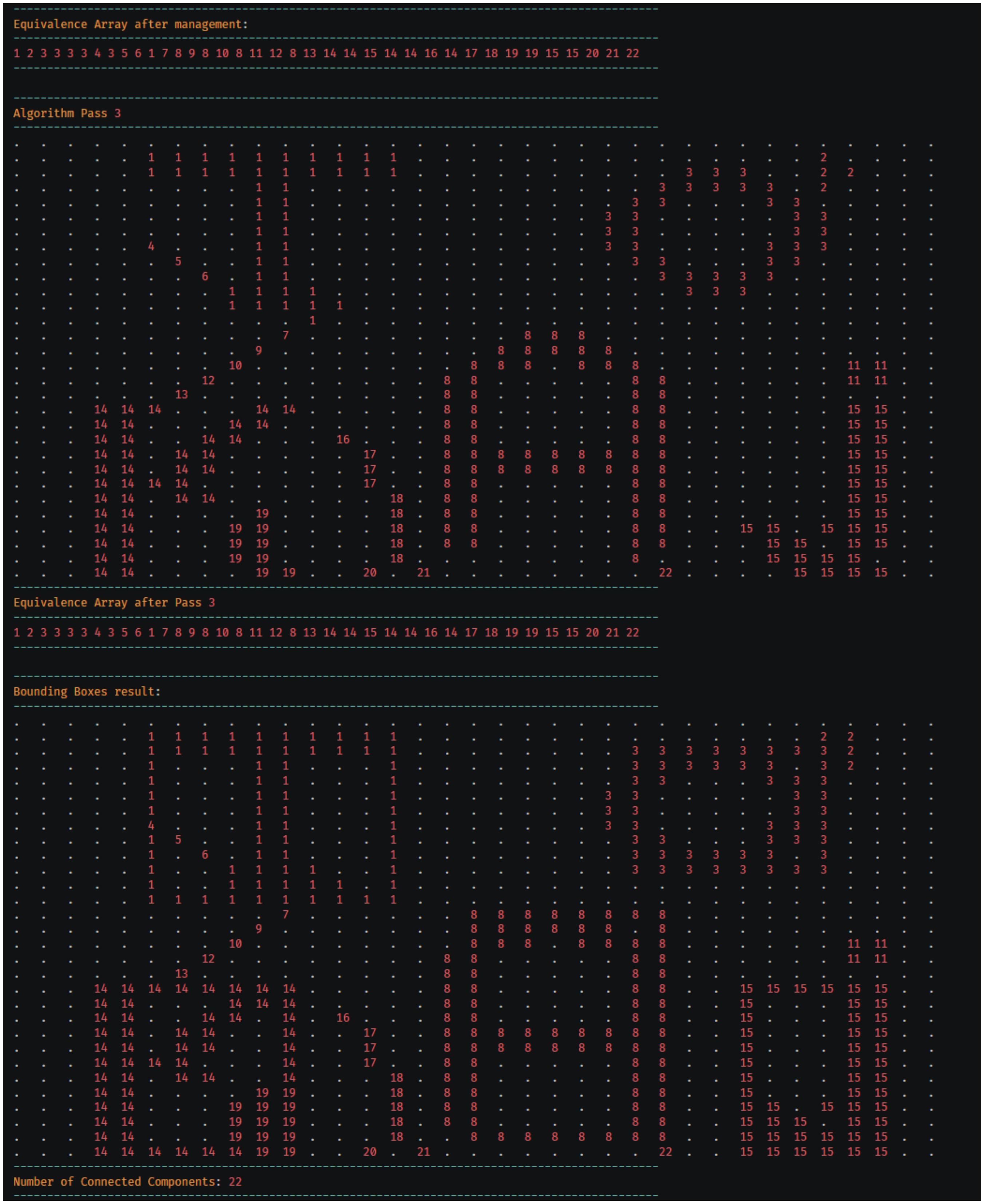


```
30 35 0 22
22
1
44
1 5
12 14
2
4
1 30
3 31
3
37
2 23
10 30
4
1
7 5
7 5
5
1
8 6
8 6
6
1
9 7
9 7
7
1
13 10
13 10
8
73
13 17
28 24
9
1
14 9
14 9
10
1
15 8
15 8
11
4
15 31
16 32
12
1
16 7
16 7
13
1
17 6
17 6
14
39
18 3
29 10
15
33
18 27
29 32
16
1
20 12
20 12
17
3
21 13
23 13
18
5
24 14
28 14
19
9
25 9
29 10
20
1
30 13
29 13
21
1
30 15
29 15
22
1
30 24
29 24
```

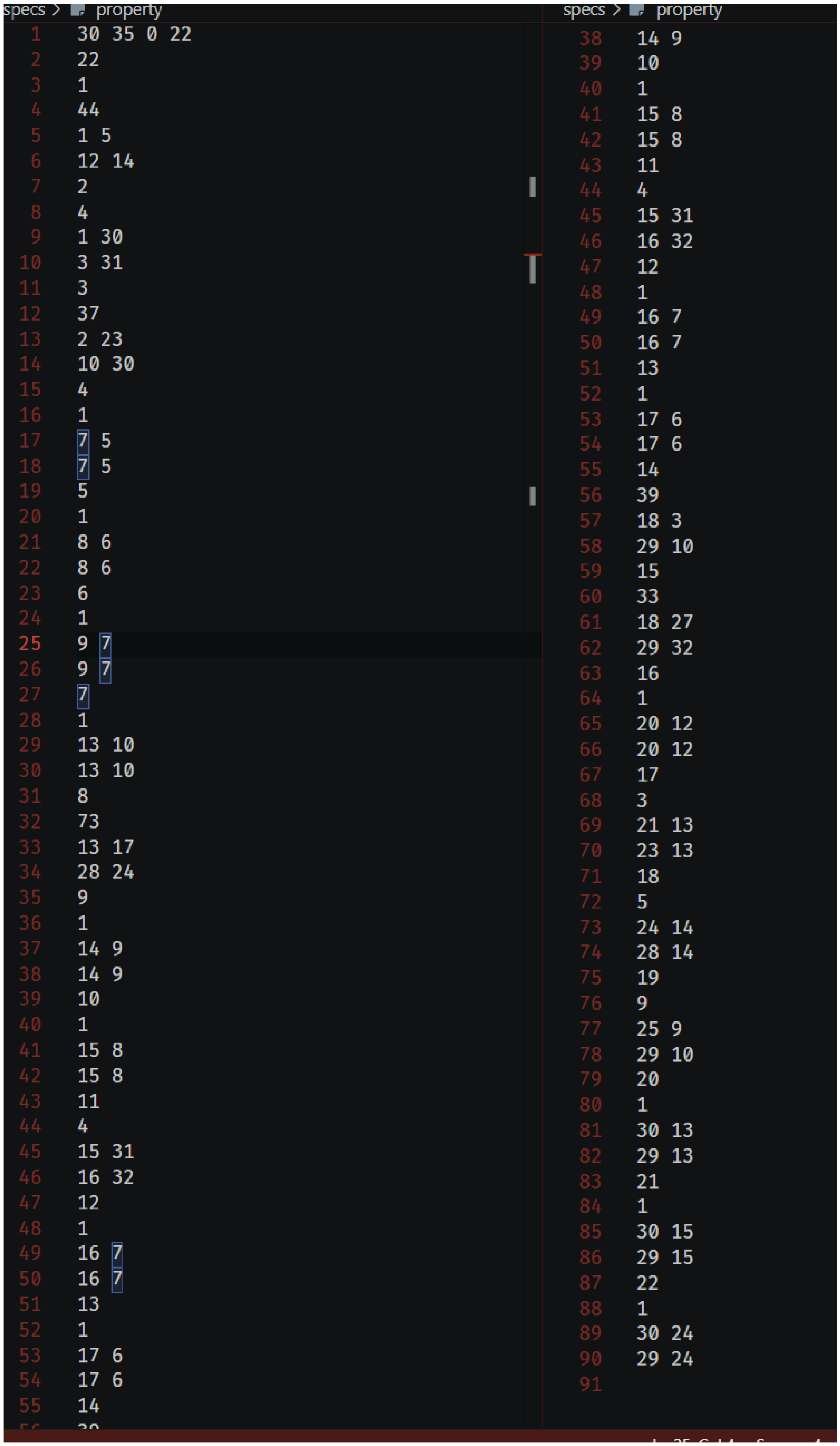
Data2 8-connectness outputs:



```
-----------------------------------------------------------------
Result of Pass 1:
-----------------------------------------------------------------
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   2   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   3   3   3   .   .   2   2   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   4   3   3   3   3   .   2   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   5   4   .   .   .   3   3   .   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   6   5   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   6   5   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   7   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   6   5   .   .   .   .   8   3   3   .   .   .   .
.   .   .   .   .   .   9   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   5   5   .   .   .   8   3   .   .   .   .   .
.   .   .   .   .   .   .   10  .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   5   5   5   5   5   .   .   .   .   .   .
.   .   .   .   .   .   .   .   11  1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   5   5   5   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   11  1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   12  .   .   .   .   .   .   .   .   13  13  13  .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   14  .   .   .   .   .   .   .   .   15  13  13  13  13  .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   16  .   .   .   .   .   .   .   .   17  15  13  .   13  13  13  .   .   .   .   .   .   .   18  18  .   .
.   .   .   .   .   .   .   19  .   .   .   .   .   .   .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   18  18  .   .
.   .   .   .   .   .   21  .   .   .   .   .   .   .   .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   .   .   .   .
.   .   .   22  22  22  .   .   .   23  23  .   .   .   .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   25  23  .   .   .   .   .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   26  25  .   .   .   27  .   .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   28  26  .   .   .   .   .   29  .   .   20  17  17  17  17  17  17  13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   28  26  .   .   .   .   .   29  .   .   20  17  17  17  17  17  17  13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  22  22  .   .   .   .   .   .   29  .   .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   22  22  .   .   .   .   .   .   30  .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   .   31  .   .   .   .   30  .   20  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   32  31  .   .   .   .   30  .   20  17  .   .   .   .   .   13  13  .   .   33  33  .   34  24  24  .   .
.   .   .   22  22  .   .   .   32  31  .   .   .   .   30  .   20  17  .   .   .   .   .   13  13  .   .   .   33  33  .   24  24  .   .
.   .   .   22  22  .   .   .   32  31  .   .   .   .   30  .   .   .   .   .   .   .   .   13  .   .   .   .   33  33  33  24  .   .   .
.   .   .   22  22  .   .   .   .   31  31  .   .   35  .   36  .   .   .   .   .   .   .   .   37  .   .   .   .   33  33  24  24  .   .
-----------------------------------------------------------------
Equivalence Array after Pass 1:
-----------------------------------------------------------------
1 2 3 4 5 5 7 5 9 10 1 12 13 14 13 16 17 18 19 17 21 22 23 24 25 26 27 22 29 30 31 31 24 24 35 36 37
-----------------------------------------------------------------


-----------------------------------------------------------------
Result of Pass 2:
-----------------------------------------------------------------
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   2   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   3   3   3   .   .   2   2   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   3   3   3   3   3   .   2   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   4   3   .   .   .   3   3   .   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   5   4   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   5   5   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   7   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   5   5   .   .   .   .   3   3   3   .   .   .   .
.   .   .   .   .   .   9   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   5   5   .   .   .   3   3   .   .   .   .   .
.   .   .   .   .   .   .   10  .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   5   5   5   5   3   .   .   .   .   .   .
.   .   .   .   .   .   .   .   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   5   5   5   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   12  .   .   .   .   .   .   .   .   13  13  13  .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   14  .   .   .   .   .   .   .   .   13  13  13  13  13  .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   16  .   .   .   .   .   .   .   .   13  13  13  .   13  13  13  .   .   .   .   .   .   .   18  18  .   .
.   .   .   .   .   .   .   19  .   .   .   .   .   .   .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   18  18  .   .
.   .   .   .   .   .   21  .   .   .   .   .   .   .   .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   .   .   .   .
.   .   .   22  22  22  .   .   .   22  22  .   .   .   .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   22  22  .   .   .   .   .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   22  22  .   .   .   27  .   .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   22  22  .   .   .   .   .   29  .   .   13  13  13  13  13  13  13  13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   22  22  .   .   .   .   .   29  .   .   13  13  13  13  13  13  13  13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  22  22  .   .   .   .   .   .   29  .   .   13  13  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   22  22  .   .   .   .   .   .   30  .   17  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   .   31  .   .   .   .   30  .   17  17  .   .   .   .   .   13  13  .   .   .   .   .   .   24  24  .   .
.   .   .   22  22  .   .   .   31  31  .   .   .   .   30  .   17  17  .   .   .   .   .   13  13  .   .   24  24  .   24  24  24  .   .
.   .   .   22  22  .   .   .   31  31  .   .   .   .   30  .   17  17  .   .   .   .   .   13  13  .   .   .   24  24  .   24  24  .   .
.   .   .   22  22  .   .   .   31  31  .   .   .   .   30  .   .   .   .   .   .   .   .   13  .   .   .   .   24  24  24  24  .   .   .
.   .   .   22  22  .   .   .   .   31  31  .   .   35  .   36  .   .   .   .   .   .   .   .   37  .   .   .   .   24  24  24  24  .   .
-----------------------------------------------------------------
Equivalence Array after Pass 2:
-----------------------------------------------------------------
1 2 3 3 4 5 7 3 9 10 1 12 13 14 13 16 13 18 19 13 21 22 22 24 22 22 27 22 29 30 31 31 24 24 35 36 37
-----------------------------------------------------------------
```



```
-----------------------------------------------------------------------------------------------
Equivalence Array after management:
-----------------------------------------------------------------------------------------------
1 2 3 3 3 3 4 3 5 6 1 7 8 9 8 10 8 11 12 8 13 14 14 15 14 14 16 14 17 18 19 19 15 15 20 21 22
-----------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------
Algorithm Pass 3
-----------------------------------------------------------------------------------------------

.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   2   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   3   3   3   .   .   2   2   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   3   3   3   3   3   .   2   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   3   3   .   .   .   3   3   .   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   3   3   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   .   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   3   3   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   4   .   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   3   3   .   .   .   .   3   3   3   .   .   .   .
.   .   .   .   .   .   5   .   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   3   3   .   .   .   3   3   .   .   .   .   .
.   .   .   .   .   .   .   6   .   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   3   3   3   3   3   .   .   .   .   .   .
.   .   .   .   .   .   .   .   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   3   3   3   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   7   .   .   .   .   .   .   .   .   8   8   8   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   9   .   .   .   .   .   .   .   .   8   8   8   8   8   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   10  .   .   .   .   .   .   .   .   8   8   8   .   8   8   8   .   .   .   .   .   .   .   11  11  .   .
.   .   .   .   .   .   .   12  .   .   .   .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   11  11  .   .
.   .   .   .   .   .   13  .   .   .   .   .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   .   .   .   .
.   .   .   14  14  14  .   .   .   14  14  .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   .   .   14  14  .   .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   .   14  14  .   .   .   16  .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   .   .   .   17  .   .   8   8   8   8   8   8   8   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   .   .   .   17  .   .   8   8   8   8   8   8   8   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  14  14  .   .   .   .   .   .   17  .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   .   .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   .   .   .   19  .   .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   15  15  .   .
.   .   .   14  14  .   .   .   19  19  .   .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   15  15  .   15  15  15  .   .
.   .   .   14  14  .   .   .   19  19  .   .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   .   15  15  .   15  15  .   .
.   .   .   14  14  .   .   .   19  19  .   .   .   .   18  .   .   .   .   .   .   .   .   8   .   .   .   .   15  15  15  15  .   .   .
.   .   .   14  14  .   .   .   .   19  19  .   .   20  .   21  .   .   .   .   .   .   .   .   22  .   .   .   .   15  15  15  15  .   .
-----------------------------------------------------------------------------------------------
Equivalence Array after Pass 3
-----------------------------------------------------------------------------------------------
1 2 3 3 3 3 4 3 5 6 1 7 8 9 8 10 8 11 12 8 13 14 14 15 14 14 16 14 17 18 19 19 15 15 20 21 22
-----------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------
Bounding Boxes result:
-----------------------------------------------------------------------------------------------

.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   2   2   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   3   3   3   3   3   3   3   3   2   .   .   .
.   .   .   .   .   1   .   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   .   3   3   3   3   3   3   .   3   2   .   .   .
.   .   .   .   .   1   .   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   .   3   3   .   .   .   3   3   3   .   .   .   .
.   .   .   .   .   1   .   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   3   3   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   1   .   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   3   3   .   .   .   .   .   3   3   .   .   .   .
.   .   .   .   .   4   .   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   3   3   .   .   .   .   3   3   3   .   .   .   .
.   .   .   .   .   1   5   .   .   1   1   .   .   .   1   .   .   .   .   .   .   .   .   3   3   .   .   .   3   3   3   .   .   .   .
.   .   .   .   .   1   .   6   .   1   1   .   .   .   1   .   .   .   .   .   .   .   .   3   3   3   3   3   3   .   3   .   .   .   .
.   .   .   .   .   1   .   .   1   1   1   1   .   .   1   .   .   .   .   .   .   .   .   3   3   3   3   3   3   3   3   .   .   .   .
.   .   .   .   .   1   .   .   1   1   1   1   1   .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   1   1   1   1   1   1   1   1   1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   .   7   .   .   .   .   .   .   8   8   8   8   8   8   8   8   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   .   9   .   .   .   .   .   .   .   8   8   8   8   8   8   .   8   .   .   .   .   .   .   .   .   .   .
.   .   .   .   .   .   .   .   10  .   .   .   .   .   .   .   .   8   8   8   .   8   8   8   8   .   .   .   .   .   .   11  11  .   .
.   .   .   .   .   .   .   12  .   .   .   .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   11  11  .   .
.   .   .   .   .   .   13  .   .   .   .   .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   .   .   .   .   .   .   .   .
.   .   .   14  14  14  14  14  14  14  14  .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   15  15  15  15  15  15  .   .
.   .   .   14  14  .   .   .   14  14  14  .   .   .   .   .   8   8   .   .   .   .   .   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   .   14  14  .   14  .   16  .   .   .   8   8   .   .   .   .   .   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   14  .   .   17  .   .   8   8   8   8   8   8   8   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   14  .   .   17  .   .   8   8   8   8   8   8   8   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  14  14  .   .   .   14  .   .   17  .   .   8   8   .   .   .   .   .   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   14  14  .   .   14  .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   .   .   .   19  19  .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   15  .   .   .   15  15  .   .
.   .   .   14  14  .   .   .   19  19  19  .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   15  15  .   15  15  15  .   .
.   .   .   14  14  .   .   .   19  19  19  .   .   .   18  .   8   8   .   .   .   .   .   8   8   .   .   15  15  15  .   15  15  .   .
.   .   .   14  14  .   .   .   19  19  19  .   .   .   18  .   .   8   8   8   8   8   8   8   8   .   .   15  15  15  15  15  15  .   .
.   .   .   14  14  14  14  14  14  19  19  .   .   20  .   21  .   .   .   .   .   .   .   .   22  .   .   15  15  15  15  15  15  .   .
-----------------------------------------------------------------------------------------------
Number of Connected Components: 22
-----------------------------------------------------------------------------------------------
```



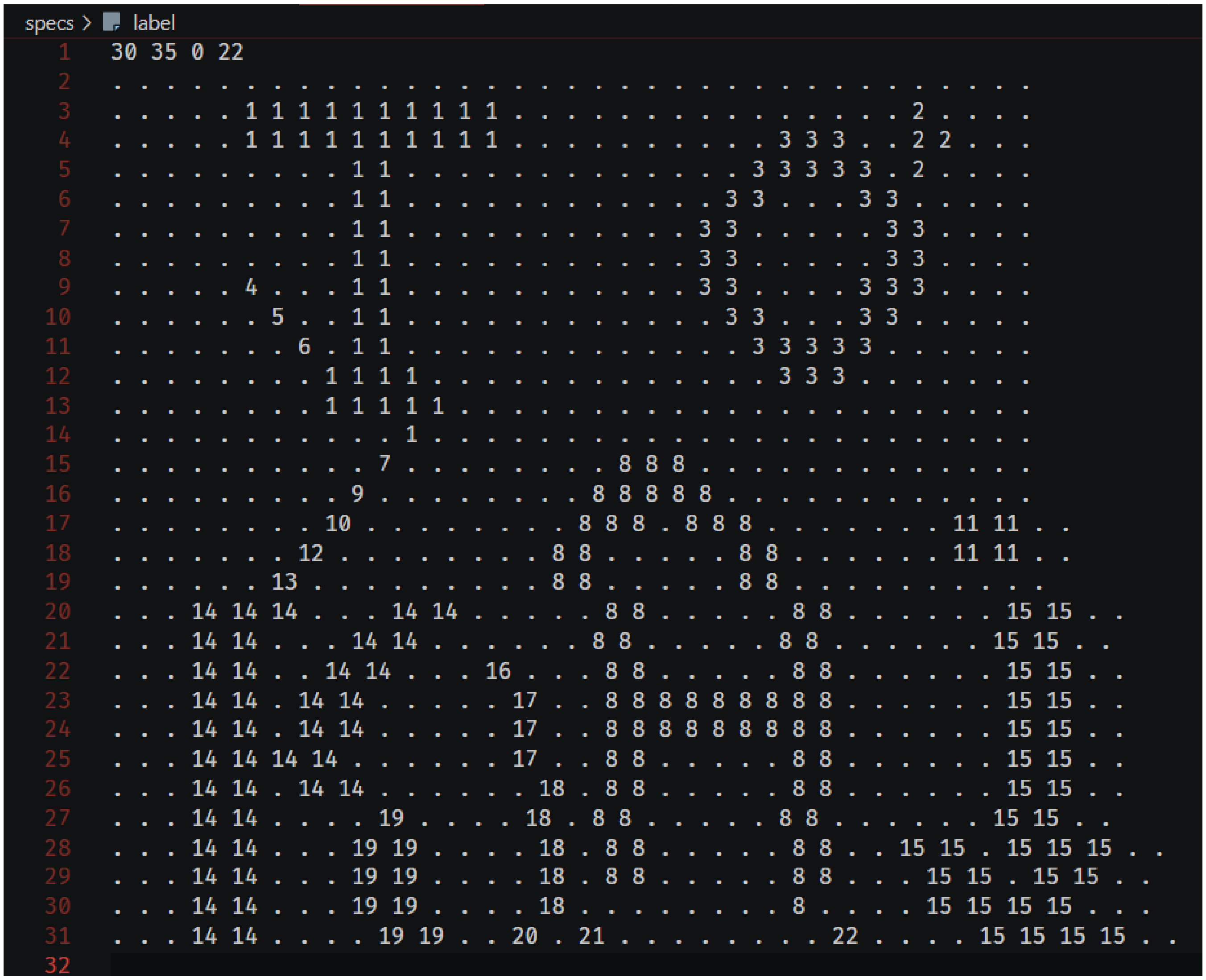
specs > property

```
30 35 0 22
22
1
44
1 5
12 14
2
4
1 30
3 31
3
37
2 23
10 30
4
1
7 5
7 5
5
1
8 6
8 6
6
1
9 7
9 7
7
1
13 10
13 10
8
73
13 17
28 24
9
1
14 9
14 9
10
1
15 8
15 8
11
4
15 31
16 32
12
1
16 7
16 7
13
1
17 6
17 6
14
```

specs > property

```
14 9
10
1
15 8
15 8
11
4
15 31
16 32
12
1
16 7
16 7
13
1
17 6
17 6
14
```

```
39
18 3
29 10
15
33
18 27
29 32
16
1
20 12
20 12
17
3
21 13
23 13
18
5
24 14
28 14
19
9
25 9
29 10
20
1
30 13
29 13
21
1
30 15
29 15
22
1
30 24
29 24
```



specs > label

```
30 35 0 22
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 . . . . . . . . . . . . . . . 2 . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 . . . . . . . . . . 3 3 3 . . 2 2 . . .
. . . . . . . . . 1 1 . . . . . . . . . . . . . 3 3 3 3 3 . 2 . . . .
. . . . . . . . . 1 1 . . . . . . . . . . . . 3 3 . . . 3 3 . . . . .
. . . . . . . . . 1 1 . . . . . . . . . . . 3 3 . . . . . 3 3 . . . .
. . . . . . . . . 1 1 . . . . . . . . . . . 3 3 . . . . . 3 3 . . . .
. . . . . 4 . . . 1 1 . . . . . . . . . . . 3 3 . . . . 3 3 3 . . . .
. . . . . . 5 . . 1 1 . . . . . . . . . . . . 3 3 . . . 3 3 . . . . .
. . . . . . . 6 . 1 1 . . . . . . . . . . . . . 3 3 3 3 3 . . . . . .
. . . . . . . . 1 1 1 1 . . . . . . . . . . . . . 3 3 3 . . . . . . .
. . . . . . . . 1 1 1 1 1 . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . 7 . . . . . . . . 8 8 8 . . . . . . . . . . . . .
. . . . . . . . . 9 . . . . . . . . 8 8 8 8 8 . . . . . . . . . . . .
. . . . . . . . 10 . . . . . . . . 8 8 8 . 8 8 8 . . . . . . . 11 11 . .
. . . . . . . 12 . . . . . . . . 8 8 . . . . . 8 8 . . . . . . 11 11 . .
. . . . . . 13 . . . . . . . . . 8 8 . . . . . 8 8 . . . . . . . . . .
. . . 14 14 14 . . . 14 14 . . . . . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . . . 14 14 . . . . . . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . . 14 14 . . . 16 . . . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . 14 14 . . . . . 17 . . 8 8 8 8 8 8 8 8 8 . . . . . . 15 15 . .
. . . 14 14 . 14 14 . . . . . 17 . . 8 8 8 8 8 8 8 8 8 . . . . . . 15 15 . .
. . . 14 14 14 14 . . . . . . 17 . . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . 14 14 . . . . . . 18 . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . . . . 19 . . . . 18 . 8 8 . . . . . 8 8 . . . . . . 15 15 . .
. . . 14 14 . . . 19 19 . . . . 18 . 8 8 . . . . . 8 8 . . 15 15 . 15 15 15 . .
. . . 14 14 . . . 19 19 . . . . 18 . 8 8 . . . . . 8 8 . . . 15 15 . 15 15 . .
. . . 14 14 . . . 19 19 . . . . 18 . . . . . . . . 8 . . . . 15 15 15 15 . . .
. . . 14 14 . . . . 19 19 . . 20 . 21 . . . . . . . . 22 . . . . 15 15 15 15 . .
```