Xinyun Wang
CV
Project#4
Connected Component
Due: 3/22/24

Algorithm Steps:

Main(…)

****

step 0: inFile open the input file from argv [1]
Connectness argv [2]
option argv [3]
RFprettyPrintFile, labelFile, propertyFile, deBugFile open from argv []
numRows, numCols, minVal, maxVal read from inFile
zeroFramedAry dynamically allocate.
newLabel 0
step 1: zero2D (zeroFramedAry)
step 2: loadImage (inFile, zeroFramedAry)
step 3: if option == ‘y’ or ‘Y’
conversion (zeroFramedAry)
step 4: if connectness == 4
connected4 (zeroFramedAry, newLabel, EQAry, RFprettyPrintFile, deBugFile)
step 5: if connectness == 8
connected8 (zeroFramedAry, newLabel, EQAry, RFprettyPrintFile, deBugFile)
step 6: labelFile output numRows, numCols, newMin, newMax to labelFile
step 7: printImg (zeroFramedAry, labelFile) // Output the result of pass3 inside of zeroFramedAry
step 8: printCCproperty (propertyFile) // print cc properties to propertyFile
step 9: drawBoxes (zeroFramedAry, CCproperty, trueNumCC) // draw on zeroFramed image.
step 10: imgReformat (zeroFramedAry, RFprettyPrintFile)
step 11: print trueNumCC to RFprettyPrintFile with proper caption
step 12: close all files

Source code:

```
#include <fstream>
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;

class Property {
public:
	int label;
	int numP;
	int minR;
	int minC;
	int maxR;
	int maxC;


	Property(int l, int np, int nr, int nc, int mr, int mc) {
		label = l;
		numP = np;
		minR = nr;
		maxR = mr;
		minC = nc;
		maxC = mc;
	}
	Property(){}
};

class ccLabel {
public:
	int numR;
	int numC;
	int minV;
	int maxV;
	int newLabel;
	int trueNumCC;
	int newMin;
	int newMax;

	int NZNAry[6];
	int** ZFAry;
	int* EQAry;
	char option;
	Property* CCproperty;

	ccLabel(ifstream& in) {
		in >> numR >> numC>> minV >> maxV;
		newLabel = 0;
```

```
	ZFAry = new int* [numR+2];
	EQAry = new int[(numR * numC)/2];
	for (int i = 0; i < numR+2; i++) {
		ZFAry[i] = new int[numC+2];
	}

}

void zero2DAry(int** ary) {
	for (int i = 0; i < numR+2; i++) {
		for (int j = 0; j < numC+2; j++) {
			ary[i][j] = 0;
		}
	}
}

void negative1D() {
	for (int i = 0; i < 6; i++) {
		NZNAry[i] = -1;
	}
}

void loadImg(ifstream& in) {
	for (int i = 1; i < numR+1; i++) {
		for (int j = 1; j < numC+1; j++) {
			in >> ZFAry[i][j];
		}
	}
}

void conversion() {
	for (int i = 1; i < numR + 1; i++) {
		for (int j = 1; j < numC + 1; j++) {
			if(ZFAry[i][j] == 1)
				ZFAry[i][j] =0;
			else
				ZFAry[i][j] = 1;
		}
	}
}

void prettyPrint(int** inAry, ofstream& out) {
	//out << numR << " " << numC << " " << minV << " " << maxV << endl;
	string s = to_string(newLabel);

	int w = s.length();
	int r = 1;
	while (r < numR + 1) {
```

```
                int c = 1;
                while (c < numC + 1) {
                        if (inAry[r][c] == 0) {
                                out << ".";
                        }
                        else {
                                out << inAry[r][c];
                        }
                        s = to_string(inAry[r][c]);
                        int ww = s.length();
                        while (ww <= w) {
                                out << " ";
                                ww++;
                        }
                        c++;
                }
                out << endl;
                r++;
        }
}

void printImg(int** inAry, ofstream& out) {
        out << numR << " " << numC << " " << minV << " " << maxV << endl;
        string s = to_string(newLabel);

        int w = s.length();
        int r = 1;
        while (r < numR + 1) {
                int c = 1;
                while (c < numC + 1) {
                        out << inAry[r][c];

                        s = to_string(inAry[r][c]);
                        int ww = s.length();
                        while (ww <= w) {
                                out << " ";
                                ww++;
                        }
                        c++;
                }
                out << endl;
                r++;
        }
}

int checkNeighbor8(int i, int j, int c) {
        negative1D();
        int n = 0;
        if(c == 1)
```

```
	{
		int x = -1;
		int y = -1;
		while (x < 1) {
			y = -1;
			while (y < 2 && !(x==0 && y>=0)) {
				if (ZFAry[i + x][j + y] > 0) {
					NZNAry[n] = ZFAry[i + x][j + y];
					n++;
				}
				y++;
			}
			x++;
		}
	}
	if (c == 2)
	{
		int x = 0;
		int y = -1;
		while(x < 2){
			if (x == 0)
				y = 0;
			else
				y = -1;
			while(y < 2){
				if (ZFAry[i + x][j + y] > 0) {
					NZNAry[n] = ZFAry[i + x][j + y];
					n++;
				}
				y++;
			}
			x++;
		}
	}
	int e = 0;
	for (int x = 1; x < n; x++) {
		if (NZNAry[x] == NZNAry[x - 1])
			e++;
	}
	NZNAry[5] = e;

	sort(NZNAry, NZNAry + n);
	return n;
}

int checkNeighbor4(int i, int j, int c) {
	negative1D();
	int n = 0;
	if (c == 1) {
```

```
                        if(ZFAry[i + -1][j + 0] > 0)
                        {
                                NZNAry[n] = ZFAry[i + -1][j + 0];
                                n++;
                        }
                        if (ZFAry[i + 0][j + -1] > 0)
                        {
                                NZNAry[n] = ZFAry[i + 0][j + -1];
                                n++;
                        }

                        if (NZNAry[0] == NZNAry[1]  && n == 2 || n == 1)
                                NZNAry[5] = 1;
                }

                if (c == 2) {
                        if (ZFAry[i + 0][j + 1] > 0)
                        {
                                NZNAry[n] = ZFAry[i + 0][j + 1];
                                n++;
                        }
                        if (ZFAry[i + 1][j + 0] > 0)
                        {
                                NZNAry[n] = ZFAry[i + 1][j + 0];
                                n++;
                        }
                        NZNAry[n] = ZFAry[i][j];
                        n++;
                        if (NZNAry[0] == NZNAry[1] && NZNAry[2] == NZNAry[1] && n == 2 || NZNAry[0]
== ZFAry[i][j] && n == 1)
                                NZNAry[5] = 1;
                }

                sort(NZNAry, NZNAry + n);
                return n;
        }

        void connect8Pass1() {
                newLabel = 0;
                for (int i = 1; i < numR + 1; i++) {
                        for (int j = 1; j < numC + 1; j++) {
                                if (ZFAry[i][j] > 0) {
                                        int n = checkNeighbor8(i, j, 1);
                                        if (n == 0) {
                                                newLabel++;
                                                ZFAry[i][j] = newLabel;
                                                EQAry[ZFAry[i][j]] = newLabel;
                                        }
                                        else if (n == (NZNAry[5] + 1) || n==1) {
```

```
                                        ZFAry[i][j] = EQAry[NZNAry[0]];
                                }
                                else {
                                        ZFAry[i][j] = NZNAry[0];
                                        updateEQ(n);
                                }
                        }
                }
        }
}

void connect8Pass2() {
        for (int i = numR+1; i > 0; i--) {
                for (int j = numC+1 ; j > 0; j--) {
                        if (ZFAry[i][j] > 0) {
                                int n = checkNeighbor8(i, j, 2);
                                if (n == 0) {
                                }
                                else if (n == (NZNAry[5] - 1)) {
                                }
                                else {
                                        int minL = EQAry[NZNAry[0]];
                                        if(ZFAry[i][j] > minL)
                                        {
                                                EQAry[ZFAry[i][j]] = minL;
                                                ZFAry[i][j] = minL;
                                                updateEQ(n);
                                        }
                                }
                                ZFAry[i][j] = EQAry[ZFAry[i][j]];
                        }
                }
        }
}

void connect4Pass1() {
        newLabel = 0;
        for (int i = 1; i < numR + 1; i++) {
                for (int j = 1; j < numC + 1; j++) {
                        if (ZFAry[i][j] > 0) {
                                int n = checkNeighbor4(i, j, 1);
                                if (n == 0) {
                                        newLabel++;
                                        ZFAry[i][j] = newLabel;
                                        EQAry[ZFAry[i][j]] = newLabel;
                                }
                                else if (1 == NZNAry[5]) {
                                        ZFAry[i][j] = EQAry[NZNAry[0]];
                                }
```

```
                                        else {
                                                ZFAry[i][j] = EQAry[NZNAry[0]];

                                                updateEQ(n);
                                        }
                                }
                        }
                }
        }

void connect4Pass2() {
        for (int i = numR + 1; i > 0; i--) {
                for (int j = numC + 1; j > 0; j--) {
                        if (ZFAry[i][j] > 0) {
                                int n = checkNeighbor4(i, j, 2);
                                if (n == 0) {
                                }
                                else if (1 == (NZNAry[5])) {
                                }
                                else {
                                        int minL = EQAry[NZNAry[0]];
                                        if (ZFAry[i][j] > minL)
                                        {
                                                EQAry[ZFAry[i][j]] = minL;
                                                ZFAry[i][j] = minL;
                                                updateEQ(n);
                                        }
                                }
                                ZFAry[i][j] = EQAry[ZFAry[i][j]];
                        }
                }
        }
}

void updateEQ(int n) {
        for (int i = 0; i < n; i++) {
                if (EQAry[NZNAry[i]] > EQAry[NZNAry[0]])
                        EQAry[NZNAry[i]] = EQAry[NZNAry[0]];
        }
}

int manageEQAry() {
        int rd = 0;
        int i = 1;
        while (i <= newLabel)
        {
                if (i != EQAry[i])
                {
                        EQAry[i] = EQAry[EQAry[i]];
```

```
                }
                else {
                        rd++;
                        EQAry[i] = rd;
                }
                i++;
        }
        return rd;
}
void connected4(ofstream& o, ofstream& debug) {
        debug << "entering connected4 method \n";
        connect4Pass1();
        debug << "After connected4 pass1, newLabel = " << newLabel << endl;
        o << "Result of: Pass 1\n";
        prettyPrint(ZFAry, o);
        o << "\nEquivalency Table after Pass 1:  ";
        printEQAry(o);

        connect4Pass2();
        debug << "After connected4 pass2, newLabel = " << newLabel << endl;
        o << "\nResult of: Pass 2\n";
        prettyPrint(ZFAry, o);
        o << "\nEquivalency Table after Pass 2:  ";
        printEQAry(o);

        trueNumCC = manageEQAry();
        o << "\nEquivalency Table after EQ Table Management:  ";
        printEQAry(o);
        newMin = 0;
        newMax = trueNumCC;
        CCproperty = new Property[trueNumCC + 1];
        debug << "In connected4, after manage EQAry, trueNumCC = " << trueNumCC << endl;

        connectPass3(CCproperty, trueNumCC, debug);
        o << "\nResult of: Pass 3\n";
        prettyPrint(ZFAry, o);
        o << "\nEquivalency Table after Pass 3:  ";
        printEQAry(o);
        debug << "Leaving connected4 method \n";
}

void connected8( ofstream& o, ofstream& debug) {
        debug << "entering connected8 method \n";
        connect8Pass1();
        debug << "After connected8 pass1, newLabel = " << newLabel << endl;
        o << "Result of: Pass 1\n";
        prettyPrint(ZFAry, o);
        o << "\nEquivalency Table after Pass 1: ";
        printEQAry(o);
```

```
	connect8Pass2();
	debug << "After connected8 pass2, newLabel = " << newLabel << endl;
	o << "\nResult of: Pass 2\n";
	prettyPrint(ZFAry, o);
	o << "\nEquivalency Table afterPass 2:  ";
	printEQAry(o);

	trueNumCC = manageEQAry();
	o << "\nEquivalency Table after EQ Table Management:  ";
	printEQAry(o);
	newMin = 0;
	newMax = trueNumCC;
	CCproperty = new Property[trueNumCC + 1];
	debug << "In connected8, after manage EQAry, trueNumCC = " << trueNumCC << endl;

	connectPass3(CCproperty, trueNumCC, debug);
	o << "\nResult of: Pass 3\n";
	prettyPrint(ZFAry, o);
	o << "\nEquivalency Table after Pass 3:  ";
	printEQAry(o);
	debug << "Leaving connected8 method \n";
}

void connectPass3(Property* cp, int tc, ofstream& debug) {
	debug << "entering connectPass3 method \n";
	for (int i = 1; i < trueNumCC + 1; i++) {
		CCproperty[i] = Property(i, 0, numR, numC, 0, 0);
	}

	for (int r = 1; r < numR + 1; r++) {
		for (int c = 1; c < numC + 1; c++) {
			if (ZFAry[r][c] > 0) {
				ZFAry[r][c] = EQAry[ZFAry[r][c]];
				int k = ZFAry[r][c];
				CCproperty[k].numP++;
				if (r < CCproperty[k].minR)
					CCproperty[k].minR = r - 1;
				if (r > CCproperty[k].maxR)
					CCproperty[k].maxR = r - 1;
				if (c <= CCproperty[k].minC)
					CCproperty[k].minC = c - 1;
				if (c > CCproperty[k].maxC)
					CCproperty[k].maxC = c - 1;
			}
		}
	}
	debug << "leaving connectPass3 method \n";
}
```

```
void printCCproperty(ofstream& o) {
	o << numR << " " << numC << " " << minV << " " << maxV << endl;
	o << trueNumCC << endl;
	for (int i = 1; i < trueNumCC + 1; i++) {
		o << CCproperty[i].label << endl;
		o << CCproperty[i].numP << endl;
		o << CCproperty[i].minR << "\t" << CCproperty[i].minC << endl;
		o << CCproperty[i].maxR << "\t" << CCproperty[i].maxC << endl;
	}
}


void drawBoxes(Property* cp, int tc, ofstream& debug) {
	debug << "entering drawBoxes method" << endl;
	int index = 1;

	while (index <= trueNumCC) {
		int minR = CCproperty[index].minR + 1;
		int minC = CCproperty[index].minC + 1;
		int maxR = CCproperty[index].maxR + 1;
		int maxC = CCproperty[index].maxC + 1;
		int label = CCproperty[index].label;

		for (int i = minC; i <= maxC; i++) {
			ZFAry[minR][i] = label;
			ZFAry[maxR][i] = label;
		}
		for (int i = minR; i <= maxR; i++) {
			ZFAry[i][minC] = label;
			ZFAry[i][maxC] = label;
		}
		index++;
	}
	debug << "leaving drawBoxes method" << endl;
}

void printEQAry(ofstream& o) {
	o << "(indexing starts from 1)" << endl;
	for (int i = 1; i <= newLabel; i++) {
		o << i << " ";
	}
	o << endl;
	for (int i = 1; i <= newLabel; i++) {
		o << EQAry[i] << " ";
		if (i >= 10 && EQAry[i]<10)
			o << " ";
	}
```

```
                o << endl;
        }

};

int main(int argc, char* argv[]) {
        ifstream in(argv[1]);
        string Connectness = (argv[2]);
        string option = (argv[3]);
        ofstream RFprettyPrintFile(argv[4]);
        ofstream labelFile(argv[5]);
        ofstream propertyFile(argv[6]);
        ofstream debug(argv[7]);
        int cn = stoi(Connectness);

        ccLabel c = ccLabel(in);
        c.zero2DAry(c.ZFAry);
        c.loadImg(in);

        if (option == "Y" || option == "y")
                c.conversion();

        if (cn == 4)
                c.connected4(RFprettyPrintFile, debug);

        if(cn == 8)
                c.connected8(RFprettyPrintFile, debug);

        c.printImg(c.ZFAry, labelFile);
        c.printCCproperty(propertyFile);
        c.drawBoxes(c.CCproperty, c.trueNumCC, debug);
        RFprettyPrintFile << "\nNumber of Connected Components: " << c.trueNumCC << endl;
        RFprettyPrintFile << "\nBounding Boxes: " << endl;
        c.prettyPrint(c.ZFAry, RFprettyPrintFile);


        labelFile.close();
        propertyFile.close();
        RFprettyPrintFile.close();
        debug.close();
        in.close();
}
```

run1 RFprettyPrintFile for 8-connectedness run on data1

```
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 . .

Equivalency Table after Pass 1: (indexing starts from 1)
1 2 3 4 5 6
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 . .

Equivalency Table afterPass 2:  (indexing starts from 1)
1 2 3 4 5 6
1 1 1 1 1 1

Equivalency Table after EQ Table Management:  (indexing starts from 1)
1 2 3 4 5 6
1 1 1 1 1 1

Result of: 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 . .

Equivalency Table after Pass 3:  (indexing starts from 1)
1 2 3 4 5 6
1 1 1 1 1 1

Number of Connected Components: 1
```

```
Bounding Boxes:
1 1 1 1 1 1 1 1 1 1
1 1 . 1 1 . 1 . 1 1
1 1 . . 1 . 1 . 1 1
1 1 . . 1 . 1 . 1 1
1 . 1 1 . . 1 . 1 1
1 . . . 1 1 1 1 1 1
1 . 1 . . . . 1 . 1
1 1 1 1 . . 1 . 1 1
1 . 1 . 1 1 1 1 . 1
1 1 1 1 1 1 1 1 1 1
```

labelFile for 8-connectedness run on data1

```
10 10 0 1
1 1 0 1 0 0 1 0 1 0
0 1 0 1 1 0 1 0 1 0
0 1 0 0 1 0 1 0 1 0
1 1 0 0 1 0 1 0 1 1
1 0 1 1 0 0 1 0 1 0
0 0 0 0 1 1 1 1 1 0
0 0 1 0 0 0 0 1 0 1
1 1 1 1 0 0 1 0 1 0
1 0 1 0 1 1 1 1 0 0
0 0 0 0 0 1 0 1 0 0
```

propertyFile for 8-connectedness run on data1

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

deBugFile

```
entering connected8 method
After connected8 pass1, newLabel = 6
After connected8 pass2, newLabel = 6
In connected8, after manage EQAry, trueNumCC = 1
entering connectPass3 method
leaving connectPass3 method
Leaving connected8 method
entering drawBoxes method
leaving drawBoxes method
```

run2 RFprettyPrintFile for 4-connectedness run on data1

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

Equivalency Table after Pass 1:  (indexing starts from 1)
1 2 3 4 5 6 7 8 9 10 11 12 13
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 .  .

Equivalency Table after Pass 2:  (indexing starts from 1)
1 2 3 4 5 6 7 8 9 10 11 12 13
1 2 3 3 1 6 3 8 9 8  11 12 11

Equivalency Table after EQ Table Management:  (indexing starts from 1)
1 2 3 4 5 6 7 8 9 10 11 12 13
1 2 3 3 1 4 3 5 6 5  7  8  7

Result of: 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  .  .

Equivalency Table after Pass 3:  (indexing starts from 1)
1 2 3 4 5 6 7 8 9 10 11 12 13
1 2 3 3 1 4 3 5 6 5  7  8  7

Number of Connected Components: 8
```

```
Bounding Boxes:
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  .  .
```

labelFile for 4-connectedness run on data1

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

propertyFile for 4-connectedness run on data1

```
10 10 0 1
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     9
6     9
7
7
7     4
9     7
8
1
7     8
7     8
```

deBugFile

```
entering connected4 method
After connected4 pass1, newLabel = 13
After connected4 pass2, newLabel = 13
In connected4, after manage EQAry, trueNumCC = 8
entering connectPass3 method
leaving connectPass3 method
Leaving connected4 method
entering drawBoxes method
leaving drawBoxes method
```

run3 RFprettyPrintFile for 4-connectedness run on data1 after conversion.

```
Result of: Pass 1
.  .  1  .  2  2  .  3  .  4
5  .  1  .  .  2  .  3  .  4
5  .  1  1  .  2  .  3  .  4
.  .  1  1  .  2  .  3  .  .
.  6  .  .  7  2  .  3  .  8
9  6  6  6  .  .  .  .  .  8
6  6  .  6  6  6  6  .  10 .
.  .  .  .  6  6  .  11 .  12
.  13 .  14 .  .  .  .  15 12
16 13 13 13 13 .  17 .  12 12

Equivalency Table after Pass 1:  (indexing starts from 1)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
1 2 3 4 5 6 2 8 6 10 11 12 13 13 12 13 17

Result of: Pass 2
.  .  1  .  2  2  .  3  .  4
5  .  1  .  .  2  .  3  .  4
5  .  1  1  .  2  .  3  .  4
.  .  1  1  .  2  .  3  .  .
.  6  .  .  2  2  .  3  .  8
6  6  6  6  .  .  .  .  .  8
6  6  .  6  6  6  6  .  10 .
.  .  .  .  6  6  .  11 .  12
.  13 .  13 .  .  .  .  12 12
13 13 13 13 13 .  17 .  12 12

Equivalency Table after Pass 2:  (indexing starts from 1)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
1 2 3 4 5 6 2 8 6 10 11 12 13 13 12 13 17

Equivalency Table after EQ Table Management:  (indexing starts from 1)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
1 2 3 4 5 6 2 7 6 8  9  10 11 11 10 11 12

Result of: Pass 3
.  .  1  .  2  2  .  3  .  4
5  .  1  .  .  2  .  3  .  4
5  .  1  1  .  2  .  3  .  4
.  .  1  1  .  2  .  3  .  .
.  6  .  .  2  2  .  3  .  7
6  6  6  6  .  .  .  .  .  7
6  6  .  6  6  6  6  .  8  .
.  .  .  .  6  6  .  9  .  10
.  11 .  11 .  .  .  .  10 10
11 11 11 11 11 .  12 .  10 10

Equivalency Table after Pass 3:  (indexing starts from 1)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
1 2 3 4 5 6 2 7 6 8  9  10 11 11 10 11 12

Number of Connected Components: 12
```

```
Bounding Boxes:
.  .  1  1  2  2  .  3  .  4
5  .  1  1  2  2  .  3  .  4
5  .  1  1  2  2  .  3  .  4
.  .  1  1  2  2  .  3  .  .
6  6  6  6  6  6  6  3  .  7
6  6  6  6  .  .  6  .  .  7
6  6  .  6  6  6  6  .  8  .
6  6  6  6  6  6  6  9  10 10
11 11 11 11 11 .  .  .  10 10
11 11 11 11 11 .  12 .  10 10
```

labelFile for 4-connectedness run on data1 after conversion.

```
10 10 0 1
0  0  1  0  2  2  0  3  0  4
5  0  1  0  0  2  0  3  0  4
5  0  1  1  0  2  0  3  0  4
0  0  1  1  0  2  0  3  0  0
0  6  0  0  2  2  0  3  0  7
6  6  6  6  0  0  0  0  0  7
6  6  0  6  6  6  6  0  8  0
0  0  0  0  6  6  0  9  0  10
0  11 0  11 0  0  0  0  10 10
11 11 11 11 11 0  12 0  10 10
```

propertyFile for 4-connectedness run on data1 after conversion.

```
10 10 0 1
12
1
6
0       2
3       3
2
7
0       4
4       5
3
5
0       7
4       7
4
3
0       9
2       9
5
2
1       0
2       0
6
13
4       0
7       6
7
2
4       9
```

```
5       9
8
1
6       8
6       8
9
1
7       7
7       7
10
5
7       8
9       9
11
7
8       0
9       4
12
1
10      6
9       6
```

deBugFile

```
entering connected4 method
After connected4 pass1, newLabel = 17
After connected4 pass2, newLabel = 17
In connected4, after manage EQAry, trueNumCC = 12
entering connectPass3 method
leaving connectPass3 method
Leaving connected4 method
entering drawBoxes method
leaving drawBoxes method
```

run4 RFprettyPrintFile for 8-connectedness run on data2

Result of: Pass 1

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

Equivalency Table after Pass 1: (indexing starts from 1)

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

Result of: Pass 2

```
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  2  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  1  .  1  .  .  .  .  .  1  1  1  .  .  .  .  .  2  .  .  .  5  .  5  .  .  .
.  7  7  .  .  .  .  .  1  1  1  .  .  .  1  1  1  1  1  .  .  .  .  .  .  .  .  .  5  5  5  .  .  .
.  7  7  .  .  .  .  .  1  .  1  1  .  .  .  1  1  1  1  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  1  1  1  .  .  .  1  1  1  1  1  1  1  .  .  .  .  8  8  8  .  .  .  .  .  .
.  .  .  .  .  .  1  1  1  1  .  1  1  1  .  .  .  .  .  .  .  .  1  1  .  .  .  .  .  .  .  .  10 10 .  .
.  .  .  .  .  1  .  .  .  1  .  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  .  .  .  .  .  10 10 .  .  .
.  .  .  .  .  1  1  1  1  1  .  .  .  .  .  .  .  .  .  1  1  1  1  1  1  1  1  .  .  .  .  .  10 .  .  .  .
.  .  .  .  .  .  .  .  1  .  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  .  .  .  .  .  .  .  .  .  .
.  .  1  .  1  .  1  .  .  1  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  1  1  .  1  .  .  1  .  .  .  .  .  .  .  1  1  1  1  1  1  1  1  1  .  .  .  .  .  .  .  .  .  .
.  .  1  .  1  .  1  .  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  .  .  .  .  .  .  .  .  .  .
.  .  1  .  1  .  .  .  .  .  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  .  .  .  1  1  .  .  1  .
.  .  1  .  .  .  .  .  .  .  .  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  .  1  .  .  .  1  .  1  1  1  .
.  .  .  1  1  .  .  .  .  .  1  1  .  1  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .  1  1  1  .
.  1  .  .  1  .  .  1  1  .  .  .  .  .  1  1  1  1  1  .  .  .  .  .  .  .  .  .  .  .  1  1  1  .
.  .  1  .  1  .  1  .  .  .  .  .  .  .  .  1  1  1  .  .  1  .  .  .  .  .  1  1  1  1  1  .
.  .  1  1  1  1  1  1  .  .  .  .  20 .  .  1  .  1  1  1  1  .  .  .  .  .  1  1  1  1  1  .
.  .  1  1  1  1  1  1  .  .  .  .  20 20 20 .  .  .  .  .  .  .  .  1  .  .  .  1  1  1  1  .  .
.  .  .  .  1  1  1  1  .  .  .  .  20 20 20 .  20 .  .  .  1  1  .  .  .  1  1  1  1  .  .  .
.  .  .  1  .  .  .  .  .  .  .  .  .  20 .  20 .  .  .  1  1  .  .  1  1  1  .  .  .  1  .
.  .  1  .  .  .  .  .  .  .  .  .  .  .  20 20 20 .  .  .  1  .  .  .  .  1  1  1  1  1  1  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  20 20 20 .  1  1  1  1  1  1  1  .  .  .  .  .  .  .
```

Equivalency Table afterPass 2: (indexing starts from 1)

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

Equivalency Table after EQ Table Management: (indexing starts from 1)

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

Result of: Pass 3

```
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . 1 . . . . . 2 . . . . . . . . . . . . .
. . . . . . . . 1 . 1 . . . . . 1 1 1 . . . . . 2 . . . 3 . 3 . . . . .
. 4 4 . . . . . 1 1 1 . . 1 1 1 1 1 . . . . . . . . 3 3 3 . . . . .
. 4 4 . . . . 1 . 1 1 . . . 1 1 1 1 1 . . . . . . . . . . . . . . . .
. . . . . . . 1 1 1 . . . 1 1 1 1 1 1 1 . . . 5 5 5 . . . . . . . . .
. . . . . 1 1 1 1 . 1 1 1 . . . . . 1 1 . . . . . 6 6 . . . . . . . .
. . . . . 1 . . 1 . 1 1 1 1 1 1 1 1 1 1 1 1 . . . . . 6 6 . . . . . .
. . . . 1 1 1 1 1 . . . . . 1 1 1 1 1 1 1 . . . . . 6 . . . . . . . .
. . . . . 1 . 1 1 1 1 1 1 1 1 1 1 1 1 1 . . . . . . . . . . . . . . .
. . 1 . 1 . 1 1 . . . . . 1 . . . . . . . . . . . . . . . . . . . . . .
. . . 1 1 . 1 . . 1 . . . . . 1 1 1 1 1 1 1 1 . . . . . . . . . . . .
. . 1 . 1 . 1 . 1 1 1 1 1 1 1 1 1 1 1 1 1 1 . . . . . . . . . . . . .
. . 1 . 1 . . . . 1 1 1 1 1 1 1 1 1 1 1 1 1 . . . 1 1 . . 1 .
. . 1 . . . . . . 1 1 1 1 1 1 1 1 1 1 1 1 . 1 . . . 1 . 1 1 .
. . . 1 1 . . . . 1 1 . 1 . . . . . 1 . . . . . . . 1 1 1 .
. 1 . . 1 . . 1 1 . . . . 1 1 1 1 1 . . . . . . . . . 1 1 1 .
. . 1 . 1 . 1 . . . . . . . 1 1 1 . . 1 . . . . . 1 1 1 1 1 .
. . 1 1 1 1 1 1 . . . . 7 . . 1 . 1 1 1 1 . . . . 1 1 1 1 1 .
. . 1 1 1 1 1 1 . . . 7 7 7 . . . . . . . 1 . . . 1 1 1 1 . .
. . . . 1 1 1 1 . . . 7 7 7 . 7 . . . 1 1 . . . 1 1 1 1 . . .
. . . 1 . . . . . . . 7 . 7 . . . 1 1 . . 1 1 1 . . . 1 .
. . 1 . . . . . . . . 7 7 7 . . . 1 . . . 1 1 1 1 1 1 .
. . . . . . . . . . . 7 7 7 . 1 1 1 1 1 1 1 . . . . . .
```

Equivalency Table after Pass 3: (indexing starts from 1)

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

Number of Connected Components: 7

Bounding Boxes:

```
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 .
. 1 . . . . . 1 . 1 . . . . . 1 1 1 . . . . . 2 . . . 3 3 3 . 1 .
. 4 4 . . . . . 1 1 1 . . 1 1 1 1 1 . . . . . . . . 3 3 3 . 1 .
. 4 4 . . . . 1 . 1 1 . . . 1 1 1 1 1 . . . . . . . . . . . . 1 .
. 1 . . . . . 1 1 1 . . . 1 1 1 1 1 1 1 . . . 5 5 5 . . . . 1 .
. 1 . . . 1 1 1 1 . 1 1 1 . . . . . 1 1 . . . . . 6 6 6 1 .
. 1 . . 1 . . 1 . 1 1 1 1 1 1 1 1 1 1 1 1 1 . . . . 6 6 6 1 .
. 1 . . 1 1 1 1 1 . . . . . 1 1 1 1 1 1 1 . . . . 6 6 6 1 .
. 1 . . . . . 1 . 1 1 1 1 1 1 1 1 1 1 1 1 1 . . . . . . . 1 .
. 1 1 . 1 . 1 1 . . . . . 1 . . . . . . . . . . . . . . 1 .
. 1 . 1 1 . 1 . . 1 . . . . . 1 1 1 1 1 1 1 1 . . . . . 1 .
. 1 1 . 1 . 1 . 1 1 1 1 1 1 1 1 1 1 1 1 1 1 . . . . . 1 .
. 1 1 . . . . . 1 1 1 1 1 1 1 1 1 1 1 1 1 . . . 1 1 . . 1 .
. 1 1 . . . . . 1 1 1 1 1 1 1 1 1 1 1 1 . 1 . . . 1 . 1 1 .
. 1 . 1 1 . . . . 1 1 . 1 . . . . . 1 . . . . . . . 1 1 1 .
. 1 . . 1 . . 1 1 . . . . 1 1 1 1 1 . . . . . . . . 1 1 1 .
. 1 1 . 1 . 1 . . . . . . . 1 1 1 . . 1 . . . . . 1 1 1 1 1 .
. 1 1 1 1 1 1 1 . . . 7 7 7 7 7 7 1 1 1 1 . . . . 1 1 1 1 1 .
. 1 1 1 1 1 1 1 . . . 7 7 7 . . 7 . . . . 1 . . . 1 1 1 1 1 .
. 1 . . 1 1 1 1 . . . 7 7 7 . 7 7 . . 1 1 . . . 1 1 1 1 . 1 .
. 1 . 1 . . . . . . . 7 . 7 . 7 7 . . 1 1 . . 1 1 1 . . . 1 .
. 1 1 . . . . . . . . 7 . . 7 7 7 . . . 1 . . . 1 1 1 1 1 1 .
. 1 1 1 1 1 1 1 1 1 1 7 7 7 7 7 7 1 1 1 1 1 1 1 1 1 1 1 1 1 .
```

labelFile for 8-connectedness run on data2

```
24 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  2  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  1  0  1  0  0  0  0  1  1  1  0  0  0  0  2  0  0  0  3  0  3  0  0  0
0  4  4  0  0  0  0  0  1  1  1  0  0  1  1  1  1  1  0  0  0  0  0  0  0  3  3  3  0  0  0
0  4  4  0  0  0  0  1  0  1  1  0  0  0  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  1  1  1  0  0  0  1  1  1  1  1  1  1  0  0  0  5  5  5  0  0  0  0  0
0  0  0  0  0  1  1  1  1  0  1  1  1  0  0  0  0  0  0  1  1  0  0  0  0  0  0  6  6  0  0
0  0  0  0  1  0  0  1  0  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  6  6  0  0  0
0  0  0  0  1  1  1  1  1  0  0  0  0  0  0  1  1  1  1  1  1  1  0  0  0  0  6  0  0  0  0
0  0  0  0  0  0  0  1  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  1  0  1  0  1  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  0  1  1  0  1  0  0  1  0  0  0  0  0  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
0  0  1  0  1  0  1  0  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  1  0  1  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  1  1  0  0  1  0
0  0  1  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1  0  1  0  0  0  1  0  1  1  0
0  0  0  1  1  0  0  0  0  1  1  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  1  1  0
0  1  0  0  1  0  0  1  1  0  0  0  0  1  1  1  1  1  0  0  0  0  0  0  0  0  0  1  1  1  0
0  0  1  0  1  0  1  0  0  0  0  0  0  0  1  1  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0
0  0  1  1  1  1  1  1  0  0  0  0  7  0  0  1  0  1  1  1  1  0  0  0  0  1  1  1  1  1  0
0  0  1  1  1  1  1  1  0  0  0  7  7  7  0  0  0  0  0  0  0  1  0  0  0  1  1  1  1  0  0
0  0  0  0  1  1  1  1  0  0  0  7  7  7  0  7  0  0  0  1  1  0  0  0  1  1  1  1  0  0  0
0  0  0  1  0  0  0  0  0  0  0  0  0  0  7  0  7  0  0  0  1  1  0  0  1  1  1  0  0  0  1  0
0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  7  7  7  0  0  0  1  0  0  0  1  1  1  1  1  1  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  7  7  0  1  1  1  1  1  1  1  0  0  0  0  0  0
```

propertyFile for 8-connectedness run on data2

```
24 31 0 1
7
1
242
1       1
23      29
2
2
1       21
2       21
3
5
2       25
3       27
4
4
3       1
4       2
5
3
5       23
5       25
6
5
6       26
8       28
7
16
18      11
23      16
```

deBugFile

```
entering connected8 method
After connected8 pass1, newLabel = 24
After connected8 pass2, newLabel = 24
In connected8, after manage EQAry, trueNumCC = 7
entering connectPass3 method
leaving connectPass3 method
Leaving connected8 method
entering drawBoxes method
leaving drawBoxes method
```

run5 RFprettyPrintFile for 4-connectedness run on data2 after conversion.

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

Equivalency Table after Pass 1:  (indexing starts from 1)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1 2 1 4 5 6 7 1 9 1  1  1  11 1  1  1  1  18 1  17 16 17 16 17 1  1  1  17 1  1  1

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  1  1  .  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  .  1  .  1  1  1  1  .  .  .  1  1  1  1  .  1  1  1  .  1  .  1  1  1
1  .  .  1  1  1  1  .  .  .  1  1  .  .  .  .  .  1  1  1  1  .  1  1  1  .  .  .  1  1  1
1  .  .  1  1  1  1  .  2  .  .  1  1  1  .  .  .  .  .  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  .  .  .  1  1  1  .  .  .  .  .  .  .  1  1  1  .  .  .  1  1  1  1  1
1  1  1  1  1  .  .  .  .  .  4  .  .  .  5  5  5  5  5  5  .  .  1  1  1  1  1  1  .  .  1  1
1  1  1  1  .  6  6  .  7  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  1  1  1  .  .  1  1  1
1  1  1  1  .  .  .  .  .  .  9  9  9  9  9  9  .  .  .  .  .  .  .  .  1  1  1  1  .  1  1  1  1
1  1  1  1  1  1  1  .  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  1  1  1  1  1  1  1  1
1  1  .  1  .  1  .  .  1  1  1  1  1  1  1  .  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  .  .  1  .  1  1  .  1  1  1  1  1  1  .  .  .  .  .  .  .  .  1  1  1  1  1  1  1  1
1  1  .  1  .  1  .  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  1  1  1  1  1  1  1
1  1  .  1  .  1  1  1  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  1  1  .  .  1  1  .  1
1  1  .  1  1  1  1  1  1  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  .  1  1  1  .  1  .  .  1
1  1  1  .  .  1  1  1  1  .  .  1  .  18 18 18 18 18 .  1  1  1  1  1  1  1  1  .  .  .  1
1  .  1  1  .  1  1  .  .  1  1  1  1  .  .  .  .  .  1  1  1  1  1  1  1  1  1  .  .  .  1
1  1  .  1  .  1  .  1  1  1  1  1  1  1  .  .  .  1  1  .  1  1  1  1  1  .  .  .  .  .  1
1  1  .  .  .  .  .  .  1  1  1  1  .  1  1  .  1  .  .  .  .  .  1  1  1  1  .  .  .  .  .  1
1  1  .  .  .  .  .  .  1  1  1  .  .  .  1  1  1  1  1  1  1  .  1  1  1  .  .  .  .  1  1
1  1  1  1  .  .  .  .  1  1  1  .  .  .  1  .  1  1  1  .  .  1  1  1  .  .  .  .  1  1  1
1  1  1  .  1  1  1  1  1  1  1  1  1  .  1  .  1  1  1  .  .  1  1  .  .  .  1  1  1  .  1
1  1  .  1  1  1  1  1  1  1  1  1  1  1  .  .  .  1  1  1  .  1  1  1  .  .  .  .  .  .  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1  .  .  .  1  .  .  .  .  .  .  .  1  1  1  1  1  1

Equivalency Table after Pass 2:  (indexing starts from 1)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1 2 1 4 5 6 7 1 9 1  1  1  1  1  1  1  1  18 1  1  1  1  1  1  1  1  1  1  1  1  1

Equivalency Table after EQ Table Management:  (indexing starts from 1)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1 2 1 3 4 5 6 1 7 1  1  1  1  1  1  1  1  8  1  1  1  1  1  1  1  1  1  1  1  1  1
```

Result of: 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  1  1  1  .  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  .  1  .  1  1  1  1  .  .  .  1  1  1  1  .  1  1  1  .  1  .  1  1
1  .  .  1  1  1  1  1  .  .  .  1  1  .  .  .  .  .  1  1  1  1  1  1  1  .  .  .  1  1
1  .  .  1  1  1  1  .  2  .  .  1  1  1  .  .  .  .  .  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  .  .  .  1  1  1  .  .  .  .  .  .  .  1  1  1  .  .  .  1  1  1  1
1  1  1  1  1  .  .  .  .  3  .  .  .  4  4  4  4  4  4  .  .  1  1  1  1  1  1  .  .  1  1
1  1  1  1  .  5  5  .  6  .  .  .  .  .  .  .  .  .  .  .  .  .  1  1  1  1  .  .  1  1  1
1  1  1  1  .  .  .  .  .  .  7  7  7  7  7  7  .  .  .  .  .  .  .  .  1  1  1  1  .  1  1  1  1
1  1  1  1  1  1  1  .  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  1  1  1  1  1  1  1  1
1  1  .  1  .  1  .  .  1  1  1  1  1  1  1  .  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  .  .  1  .  1  1  .  1  1  1  1  1  .  .  .  .  .  .  .  .  .  1  1  1  1  1  1  1  1
1  1  .  1  .  1  .  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  1  1  1  1  1  1  1
1  1  .  1  .  1  1  1  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  1  1  .  .  1  1  .  1
1  1  .  1  1  1  1  1  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  .  1  1  1  .  1  .  .  1
1  1  1  .  .  1  1  1  1  .  .  1  .  8  8  8  8  8  .  1  1  1  1  1  1  1  1  .  .  .  1
1  .  1  1  .  1  1  .  .  1  1  1  1  .  .  .  .  .  1  1  1  1  1  1  1  1  1  .  .  .  1
1  1  .  1  .  1  .  1  1  1  1  1  1  1  .  .  .  1  1  .  1  1  1  1  1  .  .  .  .  .  1
1  1  .  .  .  .  .  .  .  1  1  1  1  .  1  1  .  1  .  .  .  .  .  1  1  1  1  .  .  .  .  .  1
1  1  .  .  .  .  .  .  .  1  1  1  .  .  .  1  1  1  1  1  1  1  .  1  1  1  .  .  .  .  1  1
1  1  1  1  .  .  .  .  .  1  1  1  .  .  .  1  .  1  1  1  .  .  1  1  1  .  .  .  .  1  1  1
1  1  1  .  1  1  1  1  1  1  1  1  1  1  .  1  .  1  1  1  .  .  1  1  .  .  .  1  1  1  .  1
1  1  .  1  1  1  1  1  1  1  1  1  1  1  1  .  .  .  1  1  1  .  1  1  1  .  .  .  .  .  .  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  .  .  .  1  .  .  .  .  .  .  .  .  1  1  1  1  1  1
```

Equivalency Table after Pass 3: (indexing starts from 1)

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

Number of Connected Components: 8

Bounding Boxes:

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

labelFile for 4-connectedness run on data2 after conversion.

```
24 31 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  0  1  1  1  1  1  0  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  0  1  0  1  1  1  1  0  0  0  1  1  1  1  0  1  1  1  0  1  0  1  1  1
1  0  0  1  1  1  1  1  0  0  0  1  1  0  0  0  0  0  1  1  1  1  1  1  1  0  0  0  1  1  1
1  0  0  1  1  1  1  0  2  0  0  1  1  1  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  0  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  1  0  0  0  0  3  0  0  0  4  4  4  4  4  4  0  0  1  1  1  1  1  1  0  0  1  1
1  1  1  1  0  5  5  0  6  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  0  0  1  1  1
1  1  1  1  0  0  0  0  0  7  7  7  7  7  7  0  0  0  0  0  0  0  0  1  1  1  1  0  1  1  1  1
1  1  1  1  1  1  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1
1  1  0  1  0  1  0  0  1  1  1  1  1  1  1  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  0  0  1  0  1  1  0  1  1  1  1  1  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1
1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1
1  1  0  1  0  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  0  0  1  1  0  1
1  1  0  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  1  1  0  1  0  0  1
1  1  1  0  0  1  1  1  1  0  0  1  0  8  8  8  8  8  0  1  1  1  1  1  1  1  1  0  0  0  1
1  0  1  1  0  1  1  0  0  1  1  1  1  0  0  0  0  0  1  1  1  1  1  1  1  1  1  0  0  0  1
1  1  0  1  0  1  0  1  1  1  1  1  1  1  0  0  0  1  1  0  1  1  1  1  1  0  0  0  0  0  1
1  1  0  0  0  0  0  0  1  1  1  1  0  1  1  0  1  0  0  0  0  1  1  1  1  0  0  0  0  0  1
1  1  0  0  0  0  0  0  1  1  1  0  0  0  1  1  1  1  1  1  1  0  1  1  1  0  0  0  0  1  1
1  1  1  1  0  0  0  0  1  1  1  0  0  0  1  0  1  1  1  0  0  1  1  1  0  0  0  0  1  1  1
1  1  1  0  1  1  1  1  1  1  1  1  1  1  0  1  0  1  1  1  0  0  1  1  0  0  0  1  1  1  0  1
1  1  0  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  1  1  1  0  1  1  1  0  0  0  0  0  0  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  1  0  0  0  0  0  0  0  1  1  1  1  1  1
```

propertyFile for 4-connectedness run on data2 after conversion.

```
24 31 0 1
8
1
445
0     0
23    30
2
1
4     8
4     8
3
1
6     9
6     9
4
6
6     13
6     18
5
2
7     5
7     6
6
1
7     8
7     8
7
6
8     9
8     14
8
5
15    13
15    17
```

deBugFile

```
entering connected4 method
After connected4 pass1, newLabel = 31
After connected4 pass2, newLabel = 31
In connected4, after manage EQAry, trueNumCC = 8
entering connectPass3 method
leaving connectPass3 method
Leaving connected4 method
entering drawBoxes method
leaving drawBoxes method
```