

Guangqing Yuan

Computer Vision

Section:CSCI381-224

Project 3

**Connected Components in Binary Images**

Due date: 3/22/24

\*\*\*\*\*

#### IV. 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 <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <algorithm>
#include <limits>
using namespace std;

struct Property {
    int label, numPixels, minR, minC, maxR, maxC;
};

class ccLabel {
public:
    int numRows, numCols, minVal, maxVal, newLabel, trueNumCC, newMin, newMax;
    int** zeroFramedAry;
    int* NonZeroNeighborAry;
    int* EQAry;
    char option;
    Property* CCproperty;

    ccLabel(string inFile, char option) {
        ifstream input(inFile);
        if (!input) {
            cerr << "Error opening input file!" << endl;
            exit(1);
        }
        this->option = option;
        input >> numRows >> numCols >> minVal >> maxVal;
        zeroFramedAry = new int* [numRows + 2];
        for (int i = 0; i < numRows + 2; ++i) {
            zeroFramedAry[i] = new int[numCols + 2]();
        }
        EQAry = new int[(numRows * numCols) / 4];
        for (int i = 0; i < (numRows * numCols) / 4; ++i) {
            EQAry[i] = i;
        }
        NonZeroNeighborAry = new int[5];
        newLabel = 0;
        trueNumCC = 0;
        newMin = 0;
        newMax = 0;
        zero2D(zeroFramedAry, numRows, numCols);
        loadImage(input);
    }
};
```

```

    input.close();
}

void zero2D(int** array, int numRows, int numCols) {
    for (int i = 0; i < numRows + 2; i++) {
        fill(array[i], array[i] + numCols + 2, 0);
    }
}

void negative1D(int* array, int size) {
    fill(array, array + size, -1);
}

void loadImage(ifstream& input) {
    string line;
    int pixelValue, row = 1;

    getline(input, line);

    while (getline(input, line)) {
        stringstream ss(line);
        int col = 1;
        while (ss >> pixelValue) {
            zeroFramedAry[row][col++] = pixelValue;
        }
        row++;
    }

    if (option == 'y' || option == 'Y') {
        conversion();
    }
}

void conversion() {
    for (int i = 1; i <= numRows; i++) {
        for (int j = 1; j <= numCols; j++) {
            zeroFramedAry[i][j] = 1 - zeroFramedAry[i][j]; // Flip 0 to 1 and 1 to 0
        }
    }
}

void imgReformat(ofstream& RFprettyPrintFile, const string& caption) {
    RFprettyPrintFile << caption << endl;
    for (int i = 1; i <= numRows; i++) {
        for (int j = 1; j <= numCols; j++) {
            if (zeroFramedAry[i][j] < 10) // Single-digit numbers

```

```

        RFprettyPrintFile << zeroFramedAry[i][j] << " "; // Print with extra space
for alignment
        else
        RFprettyPrintFile << zeroFramedAry[i][j] << " "; // Print normally for
double-digit numbers
    }
    RFprettyPrintFile << endl;
}
RFprettyPrintFile << endl; // Extra line for spacing between different stages
}

```

```

void connect8Pass1() {
    newLabel = 0;
    for (int i = 1; i <= numRows; ++i) {
        for (int j = 1; j <= numCols; ++j) {
            if (zeroFramedAry[i][j] > 0) { // Foreground pixel
                // Consider 8-connectivity neighbors that are available in Pass 1
                int northWest = zeroFramedAry[i - 1][j - 1];
                int north = zeroFramedAry[i - 1][j];
                int northEast = zeroFramedAry[i - 1][j + 1];
                int west = zeroFramedAry[i][j - 1];

                // Initialize minLabel to a high value before finding the minimum
                int minLabel = numeric_limits<int>::max();
                minLabel = min(minLabel, northWest > 0 ? northWest : minLabel);
                minLabel = min(minLabel, north > 0 ? north : minLabel);
                minLabel = min(minLabel, northEast > 0 ? northEast : minLabel);
                minLabel = min(minLabel, west > 0 ? west : minLabel);

                if (minLabel == numeric_limits<int>::max()) { // No foreground neighbors
                    zeroFramedAry[i][j] = ++newLabel;
                }
                else {
                    zeroFramedAry[i][j] = minLabel;
                    // Update equivalencies for neighbors not equal to minLabel
                    if (northWest > 0 && northWest != minLabel) EQAry[northWest] =
minLabel;
                    if (north > 0 && north != minLabel) EQAry[north] = minLabel;
                    if (northEast > 0 && northEast != minLabel) EQAry[northEast] =
minLabel;
                    if (west > 0 && west != minLabel) EQAry[west] = minLabel;
                }
            }
        }
    }
}
}
}
}
}

```

```

void connect8Pass2() {
    for (int i = numRows; i > 0; --i) {
        for (int j = numCols; j > 0; --j) {
            if (zeroFramedAry[i][j] > 0) { // Check only foreground pixels
                int neighbors[] = {
                    zeroFramedAry[i][j + 1],    // East
                    zeroFramedAry[i + 1][j + 1], // South-East
                    zeroFramedAry[i + 1][j],    // South
                    zeroFramedAry[i + 1][j - 1]  // South-West
                };

                int minLabel = std::numeric_limits<int>::max();
                for (int k = 0; k < 4; ++k) {
                    if (neighbors[k] > 0) {
                        minLabel = std::min(minLabel, EQAry[neighbors[k]]);
                    }
                }

                if (minLabel < zeroFramedAry[i][j]) {
                    zeroFramedAry[i][j] = minLabel;
                    EQAry[zeroFramedAry[i][j]] = minLabel; // Update EQAry for the current
label
                }
            }
        }
    }
}

```

```

void connect4Pass1() {
    newLabel = 0; // Resetting newLabel for this pass

    for (int i = 1; i <= numRows; ++i) {
        for (int j = 1; j <= numCols; ++j) {
            if (zeroFramedAry[i][j] > 0) { // Foreground pixel
                // Examine only the North and West neighbors for 4-connectivity
                int north = zeroFramedAry[i - 1][j];
                int west = zeroFramedAry[i][j - 1];

                if (north == 0 && west == 0) { // No connected neighbors, assign a new
label
                    zeroFramedAry[i][j] = ++newLabel;
                }
                else { // One or both neighbors are foreground, assign the smallest label

```

```

        int minLabel = std::min(north > 0 ? north : newLabel, west > 0 ? west :
newLabel);
        zeroFramedAry[i][j] = minLabel;

        // Update equivalency array if necessary
        if (north > 0 && west > 0 && north != west) {
            EQAry[std::max(north, west)] = minLabel;
        }
    }
}
}
}
}
}
}
}
}
}
}

```

```

void connect4Pass2() {
    for (int i = numRows; i > 0; --i) {
        for (int j = numCols; j > 0; --j) {
            if (zeroFramedAry[i][j] > 0) { // Foreground pixel
                // Examine the South and East neighbors for 4-connectivity
                int south = zeroFramedAry[i + 1][j];
                int east = zeroFramedAry[i][j + 1];

                // Find the minimum label among the current pixel and its South and East
neighbors
                int currentLabel = zeroFramedAry[i][j];
                int minLabel = currentLabel;

                if (south > 0) {
                    minLabel = min(minLabel, EQAry[south]);
                }
                if (east > 0) {
                    minLabel = min(minLabel, EQAry[east]);
                }

                // Update the current pixel's label to the minimum label found
                zeroFramedAry[i][j] = EQAry[minLabel];

                // Update the equivalency array
                EQAry[currentLabel] = EQAry[minLabel];
            }
        }
    }
}

// At the end of Pass 2, update the equivalency array to ensure all labels point to
the smallest equivalent label
for (int label = 1; label <= newLabel; ++label) {
    EQAry[label] = EQAry[EQAry[label]];
}
}

```

```
}
```

```
void connectPass3( ) {
```

```
    for (int i = 1; i <= trueNumCC; ++i) {  
        CCproperty[i].label = i;  
        CCproperty[i].numPixels = 0;  
        CCproperty[i].minR = numRows;  
        CCproperty[i].maxR = 0;  
        CCproperty[i].minC = numCols;  
        CCproperty[i].maxC = 0;  
    }
```

```
    for (int i = 1; i <= numRows; ++i) {  
        for (int j = 1; j <= numCols; ++j) {  
            if (zeroFramedAry[i][j] > 0) {  
                int label = EQAry[zeroFramedAry[i][j]];  
                zeroFramedAry[i][j] = label;  
  
                CCproperty[label].numPixels++;  
                if (i < CCproperty[label].minR) CCproperty[label].minR = i;  
                if (i > CCproperty[label].maxR) CCproperty[label].maxR = i;  
                if (j < CCproperty[label].minC) CCproperty[label].minC = j;  
                if (j > CCproperty[label].maxC) CCproperty[label].maxC = j;  
            }  
        }  
    }
```

```
}
```

```
int manageEQAry() {  
    int trueNumCC = 0;  
    for (int i = 1; i <= newLabel; ++i) {  
        if (EQAry[i] == i) { // Root label  
            EQAry[i] = ++trueNumCC; // Assign a new component number  
        }  
        else { // Equivalent label  
            EQAry[i] = EQAry[EQAry[i]]; // Collapse to root label  
        }  
    }  
    return trueNumCC; // Return the total number of unique labels/components  
}
```



```

void printCCproperty(ofstream& propertyFile) {

    propertyFile << numRows << " " << numCols << " " << minVal << " " << maxVal <<
endl;

    propertyFile << trueNumCC << endl;

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

        propertyFile << i << endl
            << CCproperty[i].numPixels << endl
            << CCproperty[i].minR << " " << CCproperty[i].minC << endl // Upper left
corner
            << CCproperty[i].maxR << " " << CCproperty[i].maxC << endl; // Lower right
corner
        }
    }
}

```

```

void printEQAry(ofstream& outFile) {
    outFile << "Eq Table\n";
    for (int i = 1; i <= newLabel; ++i) {
        outFile << EQAry[i] << (i % 20 == 0 ? "\n" : " ");
    }
    outFile << "\n" << endl;
}

```

```

void drawBoxes() {

    for (int i = 1; i <= trueNumCC; ++i) {
        int minR = CCproperty[i].minR;
        int maxR = CCproperty[i].maxR;
        int minC = CCproperty[i].minC;
        int maxC = CCproperty[i].maxC;
        for (int row = minR; row <= maxR; ++row) {
            zeroFramedAry[row][minC] = zeroFramedAry[row][maxC] = i;
        }
        for (int col = minC; col <= maxC; ++col) {
            zeroFramedAry[minR][col] = zeroFramedAry[maxR][col] = i;
        }
    }
}

```

```

void printImg(ofstream& labelFile) {
    labelFile << numRows << " " << numCols << " " << newMin << " " << newMax <<
endl;
    for (int i = 1; i <= numRows; ++i) {
        for (int j = 1; j <= numCols; ++j) {
            labelFile << zeroFramedAry[i][j] << " ";
        }
        labelFile << endl;
    }
}

```

```

void connected4(ofstream& RFprettyPrintFile, ofstream& debugFile) {
    debugFile << "Entering connected4 method\n";

    connect4Pass1();
    debugFile << "In connected4 pass1, newLabel " << newLabel << "\n\n";
    imgReformat(RFprettyPrintFile, "After connect4Pass1");
    printEQAry(RFprettyPrintFile);
    connect4Pass2();
    debugFile << "In connected4 pass2, newLabel= " << newLabel << "\n\n";
    imgReformat(RFprettyPrintFile, "After connect4Pass2");
    printEQAry(RFprettyPrintFile);
    trueNumCC = manageEQAry();
    debugFile << "In connected4, after manage EQAry, trueNumCC=" << trueNumCC
<< "\n\n";
    connectPass3();
    imgReformat(RFprettyPrintFile, "After connectPass3");
    printEQAry(RFprettyPrintFile);
    RFprettyPrintFile << "Bounding Boexs\n";
    drawBoxes();
    printImg(RFprettyPrintFile);
    debugFile << "Leaving connected4 method\n";
}

```

```

void connected8(ofstream& RFprettyPrintFile, ofstream& debugFile) {
    RFprettyPrintFile << "HELLO\n";
    debugFile << "Entering connected8 method\n";
    connect8Pass1();
    debugFile << "In connected8 pass1, newLabel " << newLabel << "\n\n";
    imgReformat(RFprettyPrintFile, "After connect8Pass1");
    printEQAry(RFprettyPrintFile);
    connect8Pass2();
    debugFile << "In connected8 pass2, newLabel= " << newLabel << "\n\n";
    imgReformat(RFprettyPrintFile, "After connect8Pass2");
    printEQAry(RFprettyPrintFile);
}

```

```

        trueNumCC = manageEQAry();
        debugFile << "In connected8, after manage EQAry, trueNumCC=" << trueNumCC
<< "\n\n";
        manageEQAry();
        connectPass3();
        imgReformat(RFprettyPrintFile, "After connectPass3");
        printEQAry(RFprettyPrintFile);
        RFprettyPrintFile << "Bounding Boexs\n";
        drawBoxes();
        printImg(RFprettyPrintFile);
        debugFile << "Leaving connected8 method\n";
    }

};

```

```

int main(int argc, char* argv[]) {
    if (argc != 5) {
        cerr << "Usage: " << argv[0] << " <inputImage.txt> <connectness> <conversion>
<outputFilePrefix>" << endl;
        return 1;
    }

```

```

    string inputFileName = argv[1];
    int connectness = stoi(argv[2]);
    char conversion = argv[3][0]; // Assumes 'y' or 'n'
    string outputFilePrefix = argv[4];

```

```

    ofstream RFprettyPrintFile(outputFilePrefix + "_RFprettyPrint.txt");
    ofstream labelFile(outputFilePrefix + "_label.txt");
    ofstream propertyFile(outputFilePrefix + "_property.txt");
    ofstream debugFile(outputFilePrefix + "_debug.txt");

```

```

    if (!RFprettyPrintFile || !labelFile || !propertyFile || !debugFile) {
        cerr << "Error opening output files." << endl;
        return 1;
    }

```

```

    ccLabel labeler(inputFileName, conversion);

```

```

    if (connectness == 4) {
        labeler.connected4(RFprettyPrintFile, debugFile);
    }

```

```
}  
else if (connectness == 8) {  
    labeler.connected8(RFprettyPrintFile, deBugFile);  
}  
else {  
    cerr << "Invalid connectivity option. Please choose 4 or 8." << endl;  
    return 1;  
}  
  
labeler.printImg(labelFile);  
labeler.printCCproperty(propertyFile);  
labeler.printEQAry(deBugFile);  
labeler.drawBoxes();  
  
RFprettyPrintFile.close();  
labelFile.close();  
propertyFile.close();  
deBugFile.close();  
  
return 0;  
}
```

# run 1:Test and debug your program using data1 for 8-connected with option N

## RFprettyPrintFile

```
After connect8Pass1
1 1 0 2 0 0 3 0 4 0
0 1 0 2 2 0 3 0 4 0
0 1 0 0 2 0 3 0 4 0
1 1 0 0 2 0 3 0 4 4
1 0 1 1 0 0 3 0 4 0
0 0 0 0 1 1 1 1 1 0
0 0 5 0 0 0 0 1 0 1
6 5 5 5 0 0 1 0 1 0
5 0 5 0 5 1 1 1 0 0
0 0 0 0 0 1 0 1 0 0
```

```
Eq Table
1 1 1 1 1 5
```

```
After connect8Pass2
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
5 0 5 0 1 1 1 1 0 0
0 0 0 0 0 1 0 1 0 0
```

```
Eq Table
1 1 1 1 1 5
```

```
After connectPass3
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
```

```
Eq Table
1 1 1 1 1 1
```

```
Bounding Boexs
10 10 0 0
1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 0 1 0 1 1
1 1 0 0 1 0 1 0 1 1
1 1 0 0 1 0 1 0 1 1
1 0 1 1 0 0 1 0 1 1
1 0 0 0 1 1 1 1 1 1
1 0 1 0 0 0 0 1 0 1
1 1 1 1 0 0 1 0 1 1
1 0 1 0 1 1 1 1 0 1
1 1 1 1 1 1 1 1 1 1
```

## LabelFile

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

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

## propertyFile:

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

## debugFile:

```
Entering connected8 method
In connected8 pass1, newLabel 6

In connected8 pass2, newLabel= 6

In connected8, after manage EQAry, trueNumCC=1

Leaving connected8 method
Eq Table
1 1 1 1 1 1
```

**run2: Test and debug your program using data1 for 4-connected with option N until it produces the same result as given in the answer file.**

## RFpreetyPrintFile

```
After connect4Pass1
1 1 0 2 0 0 3 0 4 0
0 1 0 2 2 0 3 0 4 0
0 1 0 0 2 0 3 0 4 0
5 1 0 0 2 0 3 0 4 4
5 0 6 6 0 0 3 0 4 0
0 0 0 0 7 7 3 3 3 0
0 0 8 0 0 0 0 3 0 9
10 10 8 8 0 0 11 0 12 0
10 0 8 0 13 13 11 11 0 0
0 0 0 0 0 13 0 11 0 0
```

```
Eq Table
1 2 3 3 1 6 3 8 9 8 11 12 11
```

```
After connect4Pass2
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 6 6 0 0 3 0 3 0
0 0 0 0 3 3 3 3 3 0
0 0 8 0 0 0 0 3 0 9
8 8 8 8 0 0 11 0 12 0
8 0 8 0 11 11 11 11 0 0
0 0 0 0 0 11 0 11 0 0
```

```
Eq Table
1 2 3 3 1 6 3 8 9 8 11 12 11
```

```
After connectPass3
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
```

```
Eq Table
1 2 3 3 1 4 3 5 6 5 7 8 7
```

```
Bounding Boexs
10 10 0 0
1 1 0 2 3 3 3 3 3 3
1 1 0 2 3 0 3 0 3 3
1 1 0 2 3 0 3 0 3 3
1 1 0 2 3 0 3 0 3 3
1 1 4 4 3 0 3 0 3 3
0 0 0 0 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 0
5 5 5 5 7 7 7 7 0 0
0 0 0 0 7 7 7 7 0 0
```

## labelFile

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

## propertyFile

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

## deBugFile:

```
Entering connected4 method
In connected4 pass1, newLabel 13

In connected4 pass2, newLabel= 13

In connected4, after manage EQAry, trueNumCC=8

Leaving connected4 method
Eq Table
1 2 3 3 1 4 3 5 6 5 7 8 7
```



**run3: Test and debug your program using data1 for 4-connected with option Y. (Eyeball the result for correctness. See if you know the meaning of the result with conversion).**

## RFpreetyPrintFile

```
After connect4Pass1
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 7 2 0 3 0 8
9 6 6 6 0 0 0 0 0 8
9 6 0 6 6 6 6 0 10 0
0 0 0 0 6 6 0 11 0 12
0 13 0 14 0 0 0 0 15 12
16 13 13 13 13 0 17 0 15 12
```

```
Eq Table
1 2 3 4 5 6 2 8 6 10 11 12 13 13 12 13 17
```

```
After connect4Pass2
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 8
6 6 6 6 0 0 0 0 0 8
6 6 0 6 6 6 6 0 10 0
0 0 0 0 6 6 0 11 0 12
0 13 0 13 0 0 0 0 12 12
13 13 13 13 13 0 17 0 12 12
```

```
Eq Table
1 2 3 4 5 6 2 8 6 10 11 12 13 13 12 13 17
```

```
After connectPass3
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
```

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

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

## labelFile

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

## propertyFile

```
10 10 0 1
12
1
6
1 3
4 4
2
7
1 5
5 6
3
5
1 8
5 8
4
3
1 10
3 10
5
2
2 1
3 1
6
13
5 1
8 7
7
2
5 10
6 10
8
1
7 9
7 9
9
1
8 8
8 8
10
5
8 9
10 10
11
7
9 1
10 5
12
1
10 7
10 7
```

## deBugFile

```
Entering connected4 method
In connected4 pass1, newLabel 17

In connected4 pass2, newLabel= 17
```

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

## RFpreetyPrintFile

[illegible]

[illegible]

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

## propertyFile

```
24 31 0 1
8
1
275
2 2
24 30
2
2
2 22
3 22
3
0
24 31
0 0
4
0
24 31
0 0
5
0
24 31
0 0
6
0
24 31
0 0
7
0
24 31
0 0
8
0
24 31
0 0
```

## deBugFile

```
Entering connected8 method
In connected8 pass1, newLabel 24

In connected8 pass2, newLabel= 24

In connected8, after manage EQAry, trueNumCC=8

Leaving connected8 method
Eq Table
1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1
```

**run5: Run your program using data2 for 4-connected with option Y. (Eyeball the result for correctness. See if you know the meaning of the result with conversion)**

## RFpreetyPrintFile

[illegible]

1 1 1 0 0 1 1 1 1 0 0 17 0 18 18 18 18 18 0 19 19 16 16 1 1 1 1 0 0 0 1  
1 0 1 1 0 1 1 0 0 20 20 17 17 0 0 0 0 0 21 19 19 16 16 1 1 1 1 0 0 0 1  
1 1 0 1 0 1 0 22 22 20 20 17 17 17 0 0 0 23 21 0 19 16 16 1 1 0 0 0 0 0 1  
1 1 0 0 0 0 0 0 22 20 20 17 0 17 17 0 24 0 0 0 0 16 16 1 1 0 0 0 0 0 1  
1 1 0 0 0 0 0 0 22 20 20 0 0 0 17 17 17 17 17 17 17 0 16 1 1 0 0 0 0 25 1  
1 1 1 1 0 0 0 0 22 20 20 0 0 0 17 0 17 17 17 0 0 26 16 1 0 0 0 0 27 25 1  
1 1 1 0 28 28 28 28 22 20 20 20 20 0 17 0 17 17 17 0 0 26 16 0 0 0 29 29 27 0 1  
1 1 0 30 28 28 28 28 22 20 20 20 20 20 0 0 0 17 17 17 0 26 16 16 0 0 0 0 0 1  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 17 0 0 0 0 0 0 0 31 31 31 31 31 1

Eq Table

1 2 1 4 5 6 7 1 9 1 11 1 1 1 1 1 17 18 16 1  
19 1 21 17 1 16 25 1 27 1 1

After connect4Pass2

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 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 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 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 0 1 1 1 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 4 0 0 0 5 5 5 5 5 5 0 1 1 1 1 1 1 0 0 1 1  
1 1 1 1 0 6 6 0 7 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 9 9 9 9 9 9 0 0 0 0 0 0 1 1 1 0 1 1 1 1  
1 1 1 1 1 1 1 1 0 11 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1  
1 1 0 1 0 1 0 0 11 11 11 11 11 11 11 0 1 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 0 0 1 0 1 0 1 11 0 11 11 11 11 11 0 0 0 0 0 0 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 1 1 1 1 1 1 1 1  
1 1 0 1 0 1 1 1 1 1 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 1 1 1 0 1 0 0 1  
1 1 1 0 1 1 1 1 1 1 1 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 0 0 0 0 1 1 1 17 0 18 18 18 18 18 0 1 1 1 1 1 1 1 0 0 0 1  
1 0 1 1 0 1 1 0 0 1 1 17 17 0 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 17 17 17 0 0 0 1 19 0 1 1 1 1 1 0 0 0 0 1  
1 1 0 0 0 0 0 0 1 1 1 17 0 17 17 0 17 0 0 0 1 1 1 1 0 0 0 0 1  
1 1 0 0 0 0 0 0 1 1 1 0 0 0 17 17 17 17 17 17 17 0 1 1 1 0 0 0 0 1 1  
1 1 1 1 0 0 0 0 1 1 1 0 0 0 17 0 17 17 17 0 0 1 1 1 0 0 0 1 1 1  
1 1 1 0 1 1 1 1 1 1 1 1 1 0 17 0 17 17 17 0 1 1 0 0 1 1 25 0 1  
1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 17 17 17 0 1 1 1 0 0 0 0 0 1  
1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 17 0 0 0 0 0 0 0 1 1 1 1 1 1 1

Eq Table

1 2 1 4 5 6 7 1 9 1 11 1 1 1 1 1 17 18 1 1  
1 1 1 17 1 1 1 1 1 1

After connectPass3

1  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 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 0 1 0 1 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 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 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 1 0 5 5 0 6 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 1 1 1 1 0 1 1 1 1  
1 1 1 1 1 1 1 1 0 8 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 8 8 8 8 8 8 8 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 0 0 1 0 1 8 0 8 8 8 8 8 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 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 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 1 0 1 1 0 0 0 1  
1 1 0 1 0 1 0 1 1 1 1 1 9 9 9 9 0 0 0 1 1 1 1 1 1 1 0 0 0 1  
1 1 0 1 0 1 0 1 1 1 1 1 9 9 9 9 0 0 0 1 1 1 1 1 1 0 0 0 0 1  
1 1 0 0 0 0 0 0 1 1 1 9 0 9 9 9 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 9 9 9 9 9 9 9 9 9 0 1 1 1 0 0 0 1  
1 1 1 1 0 0 0 0 1 1 1 0 0 0 9 9 9 9 9 9 0 0 1 1 1 0 0 0 1 1 1  
1 1 1 0 1 1 1 1 1 1 1 1 1 0 9 9 9 9 9 9 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 0 0 9 9 9 9 9 9 0 1 1 1 0 0 0 0 1  
1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 9 0 0 0 0 0 0 0 1 1 1 1 1 1 1

Eq Table

1 2 1 3 4 5 6 1 7 1 8 1 1 1 1 1 9 10 1 1  
1 1 1 9 1 1 1 1 1 1 1

Bounding Boexs

24 31 0 0  
1  
1 1 1 1 1 1 1 1 1 1 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 1 1 1 0 1 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 1 0 1 1 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 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 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 1 1 1 0 1 1 1 1  
1 1 1 1 1 1 1 0 8 8 8 8 8 8 8 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1  
1 1 0 1 0 1 0 0 8 8 8 8 8 8 8 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 0 0 1 0 1 8 8 8 8 8 8 8 0 0 0 0 0 0 0 0 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 1 1 1 1 1 1 1  
1 1 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 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 1 0 1 1 1 0 1 0 0 1
1 1 1 0 0 1 1 1 1 0 0 9 9 10 10 10 10 10 9 9 9 1 1 1 1 1 1 0 0 0 1
1 0 1 1 0 1 1 0 0 1 1 9 9 0 0 0 0 0 1 1 9 1 1 1 1 1 1 0 0 0 1
1 1 0 1 0 1 0 1 1 1 1 9 9 9 0 0 0 1 1 0 9 1 1 1 1 0 0 0 0 0 1
1 1 0 0 0 0 0 0 1 1 1 9 0 9 9 0 9 0 0 0 9 1 1 1 1 0 0 0 0 0 1
1 1 0 0 0 0 0 0 1 1 1 9 0 0 9 9 9 9 9 9 9 0 1 1 1 0 0 0 0 1 1
1 1 1 1 0 0 0 0 1 1 1 9 0 0 9 0 9 9 9 0 9 1 1 1 0 0 0 0 1 1 1
1 1 1 0 1 1 1 1 1 1 1 9 1 0 9 0 9 9 9 0 9 1 1 0 0 0 1 1 1 0 1
1 1 0 1 1 1 1 1 1 1 1 9 1 1 0 0 0 9 9 9 9 1 1 1 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1 1 9 9 9 9 9 9 9 9 9 1 1 1 1 1 1 1 1 1 1
```

## labelFile

```
24 31 0 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 0 1 1 1 1 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 1 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 1 1 0 0 0 0 3 0 0 0 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 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 1 1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 0 8 8 8 8 8 8 8 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
1 1 0 1 0 1 0 0 8 8 8 8 8 8 8 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 0 0 1 0 1 8 8 8 8 8 8 8 0 0 0 0 0 0 0 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 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 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 1 0 1 1 1 0 1 0 0 1
1 1 1 0 0 1 1 1 1 1 0 0 9 9 10 10 10 10 10 9 9 9 1 1 1 1 1 0 0 0 1
1 0 1 1 0 1 1 1 0 0 1 1 9 9 0 0 0 0 0 1 1 9 1 1 1 1 1 1 0 0 0 1
1 1 0 1 0 1 0 1 1 1 1 9 9 9 0 0 0 1 1 0 9 1 1 1 1 0 0 0 0 0 1
1 1 0 0 0 0 0 0 1 1 1 9 0 9 9 0 9 0 0 0 9 1 1 1 1 0 0 0 0 0 1
1 1 0 0 0 0 0 0 1 1 1 9 0 0 9 9 9 9 9 9 9 0 1 1 1 0 0 0 0 1 1
1 1 1 1 0 0 0 0 1 1 1 9 0 0 9 0 9 9 9 0 9 1 1 1 0 0 0 0 1 1 1
1 1 1 0 1 1 1 1 1 1 1 9 1 0 9 0 9 9 9 0 9 1 1 0 0 0 1 1 1 0 1
1 1 0 1 1 1 1 1 1 1 1 9 1 1 0 0 0 9 9 9 9 1 1 1 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1 1 9 9 9 9 9 9 9 9 9 1 1 1 1 1 1 1 1 1 1
```

## propertyFile

```
24 31 0 1
10
1
402
1 1
24 31
2
1
5 9
5 9
3
1
7 10
7 10
4
6
7 14
7 19
5
2
8 6
8 7
6
1
8 9
8 9
7
6
9 10
9 15
8
14
10 9
12 15
9
29
16 12
24 21
```



```
10
5
16 14
16 18
```

## deBugFile

```
Entering connected4 method
In connected4 pass1, newLabel 31

In connected4 pass2, newLabel= 31

In connected4, after manage EQAry, trueNumCC=10

Leaving connected4 method
Eq Table
1 2 1 3 4 5 6 1 7 1 8 1 1 1 1 1 9 10 1 1
1 1 1 9 1 1 1 1 1 1 1
```