Computer Vision Programming Language: C++

Project #6 Component Connectness

Andres Quintero

Due Date:

Soft copy: 3/22/2020

Hard copy: 3/22/2020

**********Main**********

```
step 0: inFile ← open the input file
         prettyPrintFile, labelFile, propertyFile ← open from argc[]
         numRows, numCols, minVal, maxVal ← read from inFile
         dynamically allocate zeroFramedAry.

step 1: zero2D (zeroFramedAry)

step 2: loadImage(inFile, zeroFramedAry)

step 3: Connectness ← from argv[2]

step 4: newLabel ← connectPass1 (Connectness, zeroFramedAry,
NonZeroNeighborAry)

step 5: prettyPrint (prettyPrintFile)
        printEQAry (newLable, prettyPrintFile)

step 6: connectPass2 (Connectness, zeroFramedAry, NonZeroNeighborAry)

step 7: prettyPrint (prettyPrintFile
         printEQAry (newLable, prettyPrintFile)

step 8: manageEQAry (EQAry, newLabel)
         printEQAry (numCCLable, prettyPrintFile)

step 9: connectPass3 (...) // See algorithm below
        prettyPrint (prettyPrintFile
        printEQAry (numCCLable, prettyPrintFile)

step 10: output numRows, numCols, newMin, newMax to labelFile

step 11: printImg (labelFile)

step 12: printCCproperty (propertyFile

step 13: drawBoxes(zeroFramedAry, CCproperty)

step 14: prettyPrint (prettyPrintFile)

step 15: close all files
```

Source Code:

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

//DEBUGSTUFF
void printArrayDebug(int** array, int rows, int cols){
  for(int i = 0; i < rows; i++){
    for(int j = 0; j < cols; j++){
      cout << array[i][j] << " ";
    }
    cout << endl;
  }
}
struct Property{
  int label, numpixels;
  int upperLftR = INT_MAX;
  int upperLftC= INT_MAX;
  int lowerRgtR, lowerRgtC;

};

// Prototypes
void zero2DAry(int** array, int rows, int cols);
void loadImage(fstream& imgFile, int** zeroFrameAry, int row, int col);
int connectPass1(int Connectness, int** Ary, int* NonZeroNeighborAry, int rows, int cols, int*
EQArray);
int loadNonZeroPass1(int** Ary, int Connectness, int i, int j, int* NonZeroNeighborAry, bool
&diffFlag, int &nonZeroCount);
void minus1D(int* Ary1D);
void updateEQ(int* EQArray, int* NonZeroNeighborAry, int minLabel, int nonZeroCount);
void prettyPrint(int** array, fstream& outFile, int rows, int cols);
void printEQAry(int newLabel, int* EQArray, fstream& outFile);
void connectPass2(int Connectness, int** Ary, int* NonZeroNeighborAry, int rows, int cols, int*
EQArray);
int loadNonZeroPass2(int** Ary, int Connectness, int i, int j, int* NonZeroNeighborAry, bool
&diffFlag, int &nonZeroCount);
void manageEQAry(int* EQArray, int newLabel);
void connectPass3(int* EQArray, int** Ary, Property* CCproperty, int rows, int cols);
void PrintImg(int** Ary, fstream& outFile, int rows, int cols);
void printCCProperty(Property* CCproperty, fstream& outFile, int numOfCC);
void drawBoxes(int** zeroFrameAry, Property* CCproperty, int numOfCC);

int main(int argc, char* argv[]){
  //CLI inputs
  fstream inFile(argv[1]);
  // connectness integer is from argv[2]
  fstream prettyPrintFile(argv[3], fstream::out);
  fstream labelFile(argv[4], fstream::out);
  fstream propertyFile(argv[5], fstream::out);

  // Variables
  int numRows, numCols, minVal, maxVal, Connectness, newLabel, trueNumCC, numNb;
  inFile >> numRows;
  inFile >> numCols;
  inFile >> minVal;
  inFile >> maxVal;
  newLabel = 0;
  numNb = 5;
  int *NonZeroNeighborAry = new int[numNb];
  int EQSize = (numRows*numCols)/2;
  int *EQArray = new int[EQSize];

  //Setting all EQArray[i] = i
  for(int i = 0; i < EQSize; i++){
    EQArray[i] = i;
  }

  //Dynamically allocating zeroFrameAry
```

```
  int **zeroFrameAry = new int*[numRows+2];
  for(int i = 0; i < numRows+2; i++){
    zeroFrameAry[i] = new int[numCols+2];
  }
  //Step 1
  zero2DAry(zeroFrameAry, numRows+2, numCols+2);
  //Step 2
  loadImage(inFile, zeroFrameAry, numRows, numCols);
  // printArrayDebug(zeroFrameAry, numRows+2, numCols+2); //DEBUGSTUFF
  //Step 3
  Connectness = stoi(argv[2]);
  //Step 4
  newLabel = connectPass1(Connectness, zeroFrameAry, NonZeroNeighborAry, numRows, numCols,
EQArray);
  //Step 5
  prettyPrintFile << "zeroFrameAry after Pass-1: " << endl;
  prettyPrint(zeroFrameAry, prettyPrintFile, numRows+2, numCols+2); // zeroFrameAry to
prettyPrintFile
  prettyPrintFile << "EQArray after Pass-1: " << endl;
  printEQAry(newLabel, EQArray, prettyPrintFile); // print EQArray up to newLabel to pretty
  //Step 6
  connectPass2(Connectness, zeroFrameAry, NonZeroNeighborAry, numRows, numCols, EQArray);
  //Step 7
  prettyPrintFile << "zeroFrameAry after Pass-2: " << endl;
  prettyPrint(zeroFrameAry, prettyPrintFile, numRows+2, numCols+2);
  prettyPrintFile << "EQArray after Pass-2: " << endl;
  printEQAry(newLabel, EQArray, prettyPrintFile); //
  //Step 8
  manageEQAry(EQArray, newLabel);
  prettyPrintFile << "EQArray after manageEQAry: " << endl;
  printEQAry(newLabel, EQArray, prettyPrintFile);
  //Step 8.5 Creating CCproperty
  int numOfCC = -1;
  for(int i = 0; i < newLabel; i++){
    if(EQArray[i] > numOfCC){ numOfCC = EQArray[i];}
  }
  Property* CCproperty = new Property[numOfCC+1];
  //Step 9
  connectPass3(EQArray, zeroFrameAry, CCproperty, numRows, numCols);
  prettyPrintFile << "zeroFrameAry after Pass-3: " << endl;
  prettyPrint(zeroFrameAry, prettyPrintFile, numRows+2, numCols+2);
  prettyPrintFile << "EQArray after Pass-3 (Up to numOfCC): " << endl;
  printEQAry(numOfCC, EQArray, prettyPrintFile);
  //Step 9.5 Finding newMin and newMax
  int newMin = INT_MAX, newMax;
  for(int i = 1; i < numRows+1; i++){
    for(int j = 1; j < numCols+1;j++){

      if(zeroFrameAry[i][j] > newMax) {newMax = zeroFrameAry[i][j];}
      if(zeroFrameAry[i][j] < newMin) {newMin = zeroFrameAry[i][j];}
    }
  }
  //Step 10
  labelFile << numRows << " " << numCols << " " << newMin << " " << newMax << endl;
  //Step 11
  PrintImg(zeroFrameAry, labelFile, numRows, numCols);
  //Step 12
  propertyFile << numRows << " " << numCols << " " << newMin << " " << newMax << endl;
  propertyFile << numOfCC << endl;
  printCCProperty(CCproperty, propertyFile, numOfCC);
  //Step 13
  drawBoxes(zeroFrameAry, CCproperty, numOfCC);

  //Step 14
  prettyPrintFile << "zeroFrameAry after drawBoxes(): " << endl;
  prettyPrint(zeroFrameAry, prettyPrintFile, numRows+2, numCols+2);

  //Step 15
  inFile.close();
  prettyPrintFile.close();
  labelFile.close();
```

```
  propertyFile.close();
} // end of main


// Functions
void drawBoxes(int** zeroFrameAry, Property* CCproperty, int numOfCC){
  int index = 1;
  while(index < numOfCC+1){
    int minRow = CCproperty[index].upperLftR + 1;
    int minCol = CCproperty[index].upperLftC + 1;
    int maxRow = CCproperty[index].lowerRgtR + 1;
    int maxCol = CCproperty[index].lowerRgtC + 1;
    int label = CCproperty[index].label;

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

    index++;
  }
}

void printCCProperty(Property* CCproperty, fstream& outFile, int numOfCC){
  int sizeCCP = numOfCC+1;
  for(int i = 1; i <= sizeCCP; i++){
    outFile << CCproperty[i].label << endl;
    outFile << CCproperty[i].numpixels << endl;
    outFile << CCproperty[i].upperLftR << " " << CCproperty[i].upperLftC << endl;
    outFile << CCproperty[i].lowerRgtR << " " << CCproperty[i].lowerRgtC << endl;
  }

}

void PrintImg(int** Ary, fstream& outFile, int rows, int cols){
  for(int i = 1; i < rows+1; i++){
    for(int j = 1; j < cols+1; j++){
      outFile << Ary[i][j] << " ";
    }
    outFile << endl;
  }
}


void connectPass3(int* EQArray, int** Ary, Property* CCproperty, int rows, int cols){

  for(int i = 1; i < rows+1; i++){
    for(int j = 1; j < cols+1; j++){
      if(Ary[i][j] > 0){
        Ary[i][j] = EQArray[ Ary[i][j]  ];

        CCproperty[ Ary[i][j] ].label = Ary[i][j];

        CCproperty[ Ary[i][j] ].numpixels++;

        if(i < CCproperty[ Ary[i][j] ].upperLftR) {CCproperty[Ary[i][j]].upperLftR = i;} //
lowest i
        if(j < CCproperty[ Ary[i][j] ].upperLftC) {CCproperty[Ary[i][j]].upperLftC = j;} //
lowest j

        if(i > CCproperty[ Ary[i][j] ].lowerRgtR) {CCproperty[Ary[i][j]].lowerRgtR = i;} //
highest i
        if(j > CCproperty[ Ary[i][j] ].lowerRgtC) {CCproperty[Ary[i][j]].lowerRgtC = j;} //
highest j
      }
    }
  }

}
```

```
void manageEQAry(int* EQArray, int newLabel){
  //0
  int realLabel = 0;
  //1
  int index = 1;
  //2-3
  while(index <= newLabel){
    //2
    if(index != EQArray[index]){
      EQArray[index] = EQArray[EQArray[index]];
    } else {
      realLabel++;
      EQArray[index] = realLabel;
    }
    //3
    index++;
  }// 4(loop)
}

int loadNonZeroPass2(int** Ary, int Connectness, int i, int j, int* NonZeroNeighborAry, bool
&diffFlag, int &nonZeroCount){
  minus1D(NonZeroNeighborAry);
  nonZeroCount = 0;
  //1
  NonZeroNeighborAry[nonZeroCount] = Ary[i][j];
  nonZeroCount++;
  //2
  if(Ary[i+1][j] > 0){
    NonZeroNeighborAry[nonZeroCount] = Ary[i+1][j];
    nonZeroCount++;
  }
  //3
  if(Ary[i][j+1] > 0){
    NonZeroNeighborAry[nonZeroCount] = Ary[i][j+1];
    nonZeroCount++;
  }
  //4
  if(Connectness == 8){
    if(Ary[i+1][j-1] > 0){
      NonZeroNeighborAry[nonZeroCount] = Ary[i+1][j-1];
      nonZeroCount++;
    }
    if(Ary[i+1][j+1] > 0){
      NonZeroNeighborAry[nonZeroCount] = Ary[i+1][j+1];
      nonZeroCount++;
    }
  }
  //5
  int minLabel = NonZeroNeighborAry[0];
  diffFlag = false;
  //6
  int index = 1;
  // 7-8
  while(index < nonZeroCount){
    //7 (new specs)
    if(minLabel != NonZeroNeighborAry[index]){
      diffFlag = true;
    }
    if(minLabel > NonZeroNeighborAry[index]){
      minLabel = NonZeroNeighborAry[index];
    }
    //8
    index++;
  }// 9(loop)
  //10
  return minLabel;
}
```

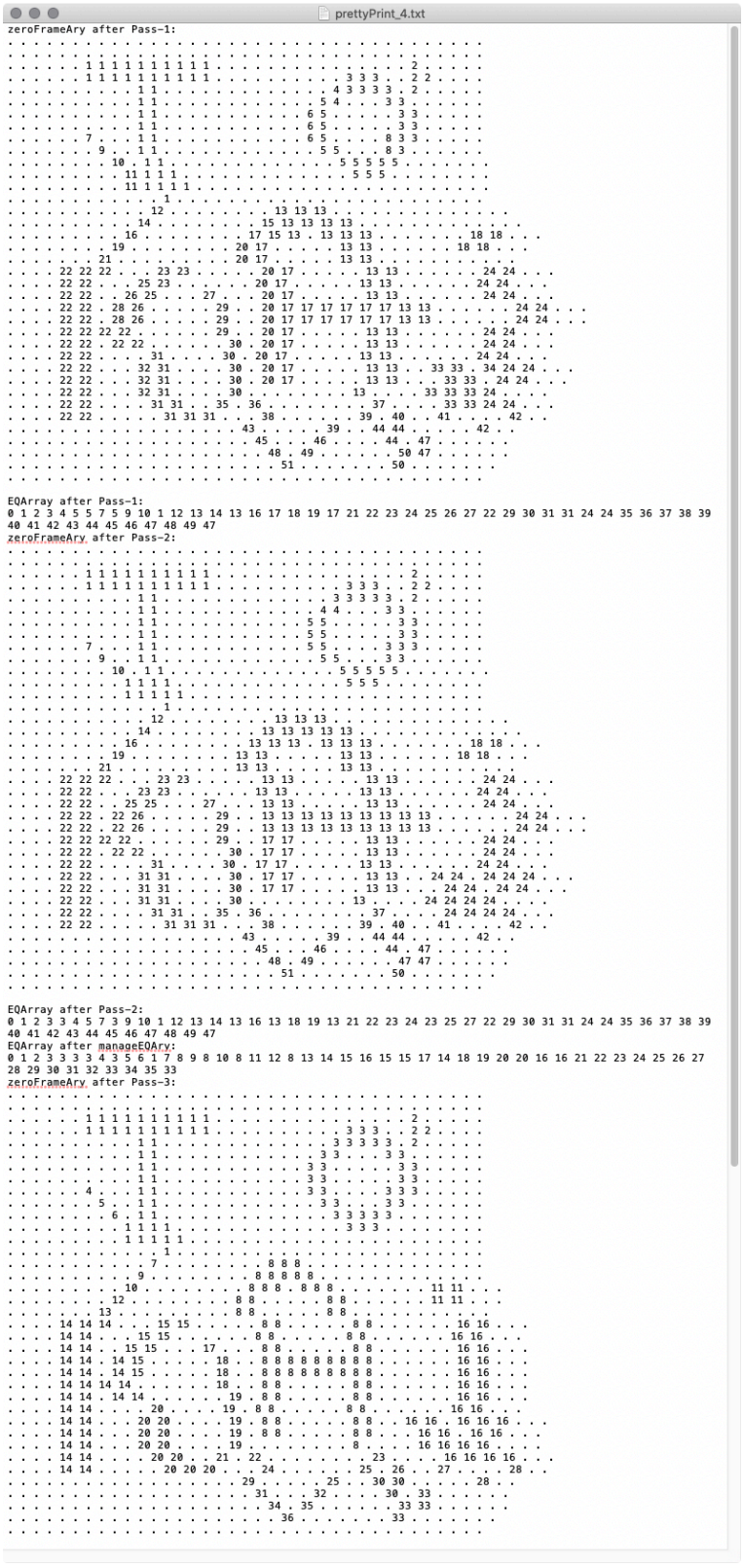
```
void connectPass2(int Connectness, int** Ary, int* NonZeroNeighborAry, int rows, int cols, int*
EQArray){
  int minLabel;
  int nonZeroCount;
  bool diffFlag;
  for(int I = rows; I > 0; i--){ //R-L
    for(int j = cols; j > 0; j--){//B-T
      if(Ary[i][j] > 0){
        minLabel = loadNonZeroPass2(Ary, Connectness, I, j, NonZeroNeighborAry, diffFlag,
nonZeroCount);
        if(minLabel != Ary[i][j]){
          Ary[i][j] = minLabel;
        }
        if(diffFlag == true){
          updateEQ(EQArray, NonZeroNeighborAry, minLabel, nonZeroCount);
        }
      }

    }
  }
}


void printEQAry(int newLabel, int* EQArray, fstream& outFile){
  for(int I = 0; I < newLabel; i++){
    outFile << EQArray[i] << " ";
  }
  outFile << endl;
}

void prettyPrint(int** array, fstream& outFile, int rows, int cols){
  for(int I = 0; I < rows; i++){
    for(int j = 0; j < cols; j++){
      if(array[i][j] == 0){
        outFile << ". ";
      } else {
        outFile << array[i][j] << " ";
      }
    }
    outFile << endl;
  }
  outFile << endl;
}

void updateEQ(int* EQArray, int* NonZeroNeighborAry, int minLabel, int nonZeroCount){
  int index = 0;
  while(index < nonZeroCount && NonZeroNeighborAry[index] != -1){
    EQArray[NonZeroNeighborAry[index]] = minLabel;
    index++;
  }
}

void minus1D(int* Ary1D){
  for(int I = 0; I < 5; i++){ //Size has been hardcoded
    Ary1D[i] = -1;
  }
}

int loadNonZeroPass1(int** Ary, int Connectness, int I, int j, int* NonZeroNeighborAry, bool
&diffFlag, int &nonZeroCount){
  minus1D(NonZeroNeighborAry);
  nonZeroCount = 0;
  //1
  if(Ary[i-1][j] > 0){
    NonZeroNeighborAry[nonZeroCount] = Ary[i-1][j];
    nonZeroCount++;
  }
  //2
  if(Ary[i][j-1] > 0){
    NonZeroNeighborAry[nonZeroCount] = Ary[i][j-1];
    nonZeroCount++;
```

```
  }
  //3
  if(Connectness == 8){
    if(Ary[i-1][j-1] > 0){
      NonZeroNeighborAry[nonZeroCount] = Ary[i-1][j-1];
      nonZeroCount++;
    }
    if(Ary[i-1][j+1] > 0){
      NonZeroNeighborAry[nonZeroCount] = Ary[i-1][j+1];
      nonZeroCount++;
    }
  }
  //4
  if(nonZeroCount <= 0){return 0;}
  //5
  int minLabel = NonZeroNeighborAry[0];
  diffFlag = false;
  //6
  int index = 1;
  //7-8
  while(index < nonZeroCount){
    //7 (new specs)
    if(minLabel != NonZeroNeighborAry[index]){
      diffFlag = true;
    }
    if(minLabel > NonZeroNeighborAry[index]){
      minLabel = NonZeroNeighborAry[index];
    }
      //8
    index++;
  }// 9(loop)
    //10
  return minLabel;

}

int connectPass1(int Connectness, int** Ary, int* NonZeroNeighborAry, int rows, int cols, int*
EQArray){
  int newLabel = 0;
  int minLabel;
  int nonZeroCount;
  bool diffFlag;
  for(int I = 1; I < rows+1; i++){
    for(int j = 1; j < cols+1; j++){
      if (Ary[i][j] > 0){
        minLabel = loadNonZeroPass1(Ary, Connectness, I, j, NonZeroNeighborAry, diffFlag,
nonZeroCount);
        if(minLabel == 0){
          newLabel++;
          Ary[i][j] = newLabel;
        } else if (minLabel > 0){
          Ary[i][j] = minLabel;
          if (diffFlag == true){
            updateEQ(EQArray, NonZeroNeighborAry, minLabel, nonZeroCount);
          }
        }
      }
    }
  } // end of loop
  return newLabel;
}

void loadImage(fstream& inFile, int** zeroFrameAry, int rows, int cols){
  int value;
  for(int I = 1; I < rows+1; i++){
    for(int j = 1; j < cols+1; j++){
      inFile >> value;
      zeroFrameAry[i][j] = value;
    }
  }
}
```

```
void zero2Dary(int** array, int rows, int cols){
  for(int I = 0; I < rows; i++){
    for(int j = 0; j < cols; j++){
      array[i][j] = 0;
    }
  }
}
```

END OF SOURCE CODE

Output for 4-Connectness:
prettyPrintFile



prettyPrint_4.txt

```
zeroFrameAry after Pass-1:
. . . . . .
. . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 . . . . . 2 . . . . .
. . . . . 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 . . .
. . . . 22 22 . . . . . 31 31 31 . . . 38 . . . . . 39 . 40 . . 41 . . . . 42 . .
. . . . . 43 . . . . . 39 . . 44 44 . . . . . 42 . .
. . . . . 45 . . . 46 . . . . 44 . 47 . . . . . .
. . . . . 48 . 49 . . . . . 50 47 . . . . . .
. . . . . 51 . . . . . 50 . . . . . .
. . . . . .

EQArray after Pass-1:
0 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 38 39
40 41 42 43 44 45 46 47 48 49 47
zeroFrameAry after Pass-2:
. . . . . .
. . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 . . . . . 2 . . . . .
. . . . . 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 4 . . . 3 3 . . . . . .
. . . . . 1 1 . . . . . 5 5 . . . . . 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 5 . . . . . .
. . . . . 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 . . . 23 23 . . . . . 13 13 . . . . . 13 13 . . . . . 24 24 . . .
. . . . 22 22 . . . 23 23 . . . . . 13 13 . . . . . 13 13 . . . . . 24 24 . . .
. . . . 22 22 . . 25 25 . . . 27 . . . 13 13 . . . . . 13 13 . . . . . 24 24 . . .
. . . . 22 22 . 22 26 . . . . . 29 . . 13 13 13 13 13 13 13 13 13 13 . . . . . 24 24 . . .
. . . . 22 22 . 22 26 . . . . . 29 . . 13 13 13 13 13 13 13 13 13 13 . . . . . 24 24 . . .
. . . . 22 22 22 22 . . . . . 29 . . 17 17 . . . . . 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 . . .
. . . . 22 22 . . . . . 31 31 31 . . . 38 . . . . . 39 . 40 . . 41 . . . . 42 . .
. . . . . 43 . . . . . 39 . . 44 44 . . . . . 42 . .
. . . . . 45 . . . 46 . . . . 44 . 47 . . . . . .
. . . . . 48 . 49 . . . . . 47 47 . . . . . .
. . . . . 51 . . . . . 50 . . . . . .
. . . . . .

EQArray after Pass-2:
0 1 2 3 3 4 5 7 3 9 10 1 12 13 14 13 16 13 18 19 13 21 22 23 24 23 25 27 22 29 30 31 31 24 24 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49 47
EQArray after manageEQAry:
0 1 2 3 3 3 3 4 3 5 6 1 7 8 9 8 10 8 11 12 8 13 14 15 16 15 15 17 14 18 19 20 20 16 16 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 33
zeroFrameAry after Pass-3:
. . . . . .
. . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 . . . . . 2 . . . . .
. . . . . 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 . . . . . .
. . . . . 10 . . . . . 8 8 8 . 8 8 8 . . . . . 11 11 . . .
. . . . . 12 . . . . . 8 8 . . . . . 8 8 . . . . . 11 11 . . .
. . . . . 13 . . . . . 8 8 . . . . . 8 8 . . . . . .
. . . . 14 14 14 . . . 15 15 . . . . . 8 8 . . . . . 8 8 . . . . . 16 16 . . .
. . . . 14 14 . . . 15 15 . . . . . 8 8 . . . . . 8 8 . . . . . 16 16 . . .
. . . . 14 14 . . 15 15 . . . 17 . . . 8 8 . . . . . 8 8 . . . . . 16 16 . . .
. . . . 14 14 . 14 15 . . . . . 18 . . 8 8 8 8 8 8 8 8 8 . . . . . 16 16 . . .
. . . . 14 14 . 14 15 . . . . . 18 . . 8 8 8 8 8 8 8 8 8 . . . . . 16 16 . . .
. . . . 14 14 14 14 . . . . . 18 . . 8 8 . . . . . 8 8 . . . . . 16 16 . . .
. . . . 14 14 . 14 14 . . . . . 19 . 8 8 . . . . . 8 8 . . . . . 16 16 . . .
. . . . 14 14 . . . . 20 . . . . 19 . 8 8 . . . . . 8 8 . . . . . 16 16 . . .
. . . . 14 14 . . . 20 20 . . . . 19 . 8 8 . . . . . 8 8 . . 16 16 . 16 16 16 . . .
. . . . 14 14 . . . 20 20 . . . . 19 . 8 8 . . . . . 8 8 . . . 16 16 . 16 16 . . .
. . . . 14 14 . . . 20 20 . . . . 19 . . . . . 8 . . . . 16 16 16 16 . . . .
. . . . 14 14 . . . . 20 20 . . 21 . 22 . . . . . 23 . . . . 16 16 16 16 . . .
. . . . 14 14 . . . . . 20 20 20 . . . 24 . . . . . 25 . 26 . . 27 . . . . 28 . .
. . . . . 29 . . . . . 25 . . 30 30 . . . . . 28 . .
. . . . . 31 . . . 32 . . . . . 30 . 33 . . . . . .
. . . . . 34 . 35 . . . . . 33 33 . . . . . .
. . . . . 36 . . . . . 33 . . . . . .
. . . . . .
```

prettyPrint_4.txt

```
EQArray after Pass-3 (Up to numOfCC):
0 1 2 3 3 3 3 4 3 5 6 1 7 8 9 8 10 8 11 12 8 13 14 15 16 15 15 17 14 18 19 20 20 16 16
zeroFrameAry after drawBoxes():
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . 1 1 1 1 1 1 1 1 1 1 . . . . . . . . . . . . . . . 2 . . . . .
. . . . . . 1 1 1 1 1 1 1 1 1 1 1 . . . . . . . . . 3 3 3 . . 2 2 2 . . .
. . . . . . . 1 . . 1 1 . . . . 1 . . . . . . . 3 3 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 3 . . . .
. . . . . . . 1 . . 1 1 . . . . 1 . . . . . . 3 3 . . . . . 3 3 3 . . . .
. . . . . . 4 1 . . 1 1 . . . . 1 . . . . . . 3 3 . . . . 3 3 3 3 . . . .
. . . . . . . 1 . . 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 . . . .
. . . . . . . 1 . 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 . . . . 8 8 8 . . . . . . . . . . . . . .
. . . . . . . . . . 9 . . . . . . . 8 8 8 8 8 8 8 8 8 . . . . . . . . . .
. . . . . . . . . 10 . . . . . . . . 8 8 8 . 8 8 8 . 8 . . . . . 11 11 . . .
. . . . . . . . 12 . . . . . . . . 8 8 . . . . . 8 8 8 . . . . . 11 11 11 . .
. . . . . . . 13 . . . . . . . . . 8 8 . . . . . 8 8 8 . . . . . . 11 . . .
. . . . 14 14 14 . . . 15 15 . . . . . 8 8 . . . . . 8 8 8 . . . . . 16 16 . . .
. . . . 14 14 14 14 14 15 15 15 15 . . . . 8 8 . . . . . 8 8 8 . . 16 16 16 16 16 16 . .
. . . . 14 14 . . 15 15 . . 15 17 . . . 8 8 . . . . . 8 8 8 . . 16 . . 16 16 16 . .
. . . . 14 14 . 14 15 15 . . 15 . 18 . . 8 8 8 8 8 8 8 8 8 8 . . 16 . . 16 16 16 . .
. . . . 14 14 . 14 15 15 . . 15 . 18 18 . 8 8 8 8 8 8 8 8 8 8 . . 16 . . 16 16 16 . .
. . . . 14 14 14 14 . 15 15 15 . . 18 18 . 8 8 . . . . . 8 8 8 . . 16 . . 16 16 16 . .
. . . . 14 14 . 14 14 14 . . . . . 19 . 8 8 . . . . . 8 8 8 . . 16 . . 16 16 16 . .
. . . . 14 14 . . . 14 20 . . . . 19 19 8 8 . . . . . 8 8 8 . . 16 . . 16 16 16 . .
. . . . 14 14 . . . 14 20 20 20 20 20 19 19 8 8 . . . . . 8 8 8 . 16 16 . 16 16 16 16 . .
. . . . 14 14 . . . 14 20 . . . 20 19 19 8 8 . . . . . 8 8 8 . . 16 16 . 16 16 16 . .
. . . . 14 14 . . . 14 20 . . . 20 19 19 . 8 . . . . . 8 . 8 . . 16 16 16 16 . 16 . .
. . . . 14 14 . . . 14 20 20 . . 20 . 22 . 8 8 8 8 8 8 8 8 . . . 16 16 16 16 16 16 . .
. . . . 14 14 . . . 14 20 20 20 20 20 . . 24 . . . . . . 25 . 26 . . 16 16 16 16 16 28 . .
. . . . . 14 14 14 14 . 20 20 20 20 . . . . 29 . . . . . 25 25 . 30 30 . . . . . 28 28 .
. . . . . . . . . . . . . . . . . . . 31 . . . 32 . . . . 30 30 33 . . . . . .
. . . . . . . . . . . . . . . . . . . . 34 . 35 . . . . . 30 33 33 33 . . . . .
. . . . . . . . . . . . . . . . . . . . . 36 . . . . . . . 33 33 33 . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33 . . . . . .
```

END OF PRETTY_4

labelFile

labelFile_4.txt

```
35 35 0 36
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 3 3 3 0 0 2 2 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 3 3 3 0 2 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 3 3 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0
0 0 0 0 0 4 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 0 3 3 3 0 0 0 0
0 0 0 0 0 0 5 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 3 3 0 0 0 0 0
0 0 0 0 0 0 0 6 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 3 3 3 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 8 8 8 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 9 0 0 0 0 0 0 0 0 8 8 8 8 8 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0 8 8 8 0 8 8 8 0 0 0 0 0 0 0 11 11 0 0
0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 11 11 0 0
0 0 0 0 0 0 13 0 0 0 0 0 0 0 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 0 0 0 0
0 0 0 14 14 14 0 0 0 15 15 0 0 0 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 0 0 15 15 0 0 0 0 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 0 15 15 0 0 0 17 0 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 14 15 0 0 0 0 0 18 0 0 8 8 8 8 8 8 8 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 14 15 0 0 0 0 0 18 0 0 8 8 8 8 8 8 8 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 14 14 0 0 0 0 0 0 18 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 14 14 0 0 0 0 0 0 19 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 0 0 0 20 0 0 0 0 19 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 0 0 20 20 0 0 0 0 19 0 8 8 0 0 0 0 0 8 8 0 0 16 16 0 16 16 16 0 0
0 0 0 14 14 0 0 0 20 20 0 0 0 0 19 0 8 8 0 0 0 0 0 8 8 0 0 0 16 16 0 16 16 0 0
0 0 0 14 14 0 0 0 20 20 0 0 0 0 19 0 0 0 0 0 0 0 0 8 0 0 0 0 16 16 16 16 0 0 0
0 0 0 14 14 0 0 0 0 20 20 0 0 21 0 22 0 0 0 0 0 0 0 0 23 0 0 0 0 16 16 16 16 0 0
0 0 0 14 14 0 0 0 0 0 20 20 20 0 0 0 24 0 0 0 0 0 0 25 0 26 0 0 27 0 0 0 0 28 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 29 0 0 0 0 0 25 0 0 30 30 0 0 0 0 0 28 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 31 0 0 0 32 0 0 0 0 30 0 33 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 34 0 35 0 0 0 0 0 0 33 33 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 36 0 0 0 0 0 0 0 33 0 0 0 0 0 0
```

END OF LABEL_4

propertyFile

propertyFile_4.txt

```
35 35 0 36
35
1
44
2 6
13 15
2
4
2 31
4 32
3
37
3 23
11 31
4
1
8 6
8 6
5
1
9 7
9 7
6
1
10 8
10 8
7
1
14 11
14 11
8
73
14 17
29 25
9
1
15 10
15 10
10
1
16 9
16 9
11
4
16 32
17 33
12
1
17 8
17 8
13
1
18 7
18 7
14
33
19 4
31 8
15
8
19 8
23 11
16
33
19 28
30 33
17
1
21 13
21 13
18
3
22 14
24 14
19
5
25 15
29 15
20
12
26 9
31 13
21
1
30 14
30 14
22
1
30 16
30 16
23
1
30 25
30 25
24
1
31 17
31 17
25
2
31 24
32 24
26
1
31 26
31 26
27
1
31 29
31 29
28
2
31 34
32 34
29
1
32 18
32 18
```

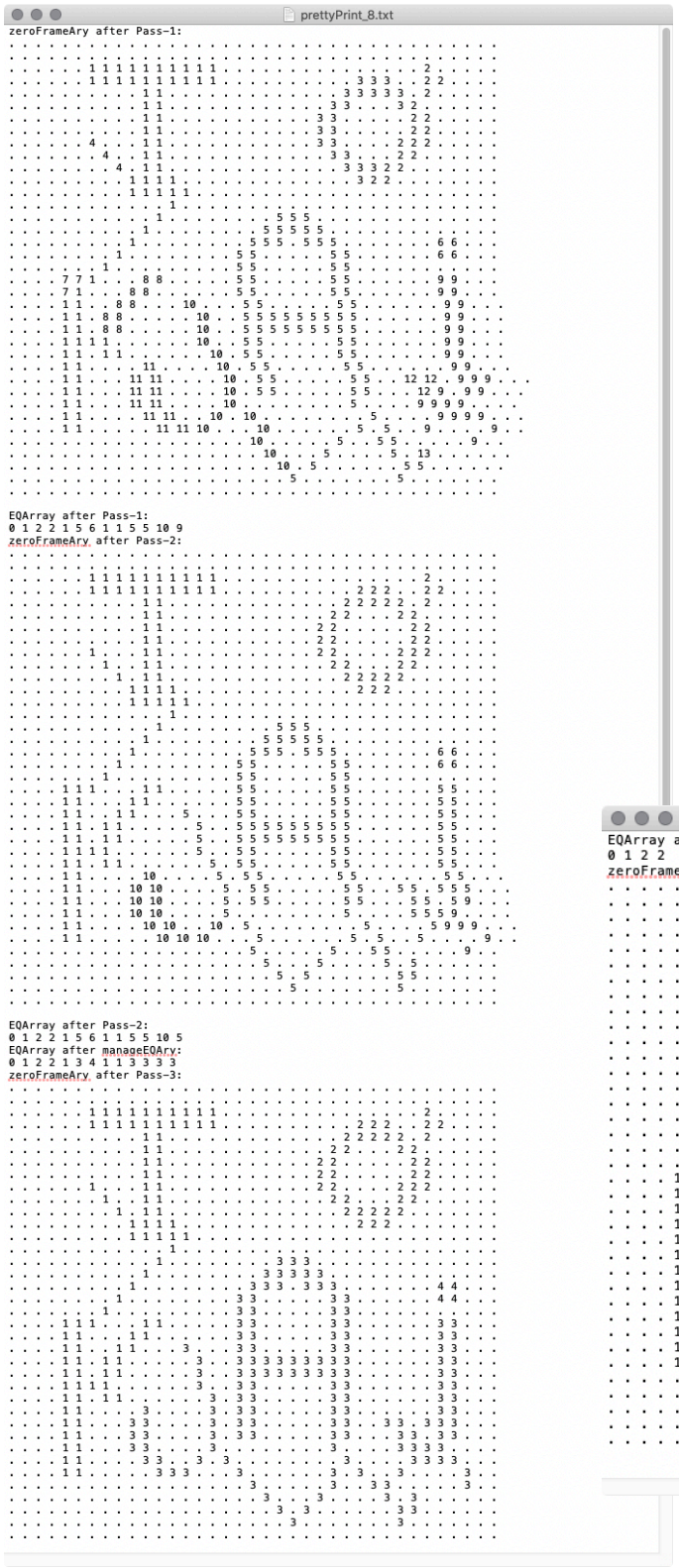
propertyFile_4.txt

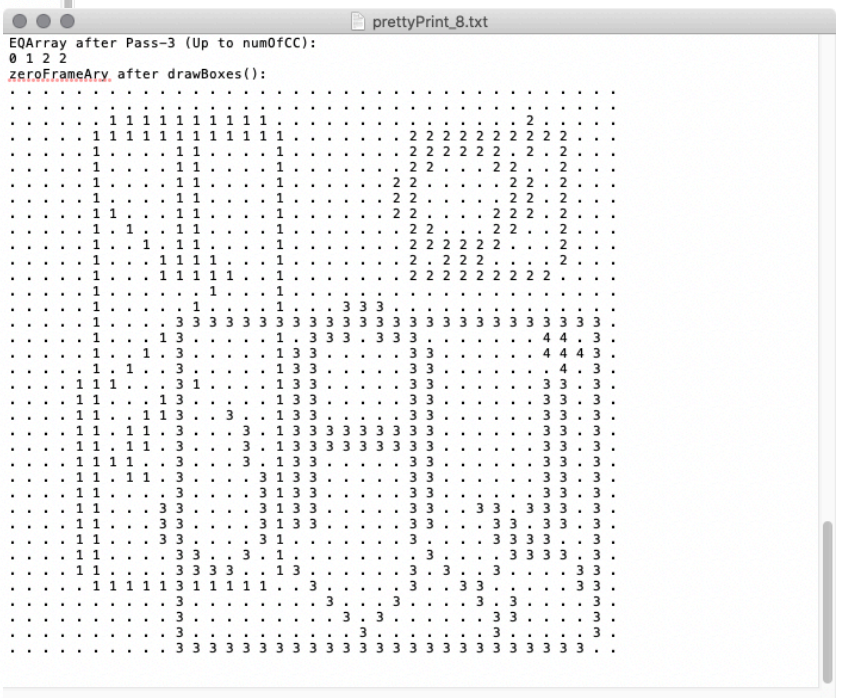
```
30
3
32 27
33 28
31
1
33 19
33 19
32
1
33 23
33 23
33
4
33 29
35 30
34
1
34 20
34 20
35
1
34 22
34 22
36
1
0 0
35 21
```

END OF PROPERTY_4

END OF CONNECTNESS 4 OUTPUT

Output for 4-Connectness:
prettyPrintFile





END OF PRETTY_8

labelFile

labelFile_8.txt

```
35 35 0 4
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 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 0 0 2 2 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 0 2 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 2 2 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 0 2 2 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 0 2 2 0 0 0
0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 2 2 2 0 0 0
0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 2 2 0 0 0 0
0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 3 3 0 3 3 3 0 0 0 0 0 0 0 4 4 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 0 4 4 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 0 3 3 0 0
0 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 0 3 3 0 0
0 0 0 1 1 0 0 1 1 0 0 0 3 0 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 3 3 0 0
0 0 0 1 1 0 1 1 0 0 0 0 0 3 0 0 3 3 3 3 3 3 3 3 3 0 0 0 0 0 0 3 3 0 0
0 0 0 1 1 0 1 1 0 0 0 0 0 3 0 0 3 3 3 3 3 3 3 3 3 0 0 0 0 0 0 3 3 0 0
0 0 0 1 1 1 0 0 0 0 0 0 3 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 0 3 3 0 0
0 0 0 1 1 0 1 1 0 0 0 0 0 3 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 0 3 3 0 0
0 0 0 1 1 0 0 0 0 3 0 0 0 0 3 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 3 3 0 0
0 0 0 1 1 0 0 0 3 3 0 0 0 0 3 0 3 3 0 0 0 0 0 3 3 0 0 0 3 3 0 3 3 0 0
0 0 0 1 1 0 0 0 3 3 0 0 0 0 3 0 3 3 0 0 0 0 0 3 3 0 0 0 3 3 0 3 3 0 0
0 0 0 1 1 0 0 0 3 3 0 0 0 0 3 0 0 0 0 0 0 0 0 3 0 0 0 0 3 3 3 3 0 0 0
0 0 0 1 1 0 0 0 0 3 3 0 0 0 3 0 3 0 0 0 0 0 0 3 0 0 0 0 3 3 3 3 0 0 0
0 0 0 1 1 0 0 0 0 0 3 3 0 0 3 0 0 0 0 0 0 0 0 3 0 3 0 0 3 0 0 0 0 3 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 3 3 0 0 0 0 0 3 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 3 0 0 0 3 0 3 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 3 0 3 0 0 0 0 0 0 3 3 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 3 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0
```

END OF LABEL_8

propertyFile

propertyFile_8.txt

```
35 35 0 4
4
1
93
2 4
31 15
2
41
2 23
11 32
3
150
14 9
35 34
4
4
16 32
17 33
0
0
0 0
0 0
```

END OF PROPERTY_8
END OF CONNECTNESS 8 OUTPUT