323.25 Project 6: Quad Tree

Student: Jingshi Liu

Due Date: 4/6/2022

Algorithm Steps:

step 0: inFile, outFile1, outFile2←open

numRows, numCols, minVal, maxVal←read from inFile
power2←computePower2(numRows, numCols)
outFile2←output power2 to outFile2 with caption imgAry←dynamically allocate the array size of power2 by power2Size zero2DAry (imgAry) // May not need it.

step 1: loadImage (inFile, imgAry)
step 2: outFile2←output imgAry to outFile2 with caption
step 3: QtRoot←BuildQuadTree (imgAry, 0, 0, power2, outFile2) step 4: preOrder (QtRoot, outFile1)
step 5: postOrder (QtRoot, outFile1)
step 6: close all files

Note: Outfile 2 for both images are attached in the email for better readability.

Source Code:

```
package com.company;


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


class QuadTreeNode{
    int color, upperRow, upperCol, size;
    QuadTreeNode NW, NE, SW, SE;


    QuadTreeNode(int upperRow, int upperCol, int size,  QuadTreeNode NW,
QuadTreeNode NE, QuadTreeNode SW, QuadTreeNode SE){
        this.upperRow = upperRow;
        this.upperCol = upperCol;
        this.size = size;
        this.NW = NW;
        this.NE = NE;
        this.SW = SW;
        this.SE = SE;
    }


    void printQuadTreeNode(BufferedWriter outFile) throws IOException {
        String res = color + ", " + upperRow + ", " + upperCol + ", ";
        if(NW == null) res += "null"; else res += NW.color;
        if(NE == null) res += "null"; else res += NE.color;
        if(SW == null) res += "null"; else res += SW.color;
        if(SE == null) res += "null"; else res += SE.color;
        outFile.write(res + "\n");
    }
```

```
}

class QuadTree{
    int numRows, numCols, minVal, maxVal, power2;
    int[][] imgArray;
    QuadTreeNode root;

    QuadTree(int numRows, int numCols, int minVal, int maxVal){
        this.numRows = numRows;
        this.numCols = numCols;
        this.minVal = minVal;
        this.maxVal = maxVal;
        imgArray = new int[computePower2()][computePower2()];
    }

    int computePower2(){
        int size = Math.max(numCols, numRows), power2 = 2;

        while(size > power2)
            power2 *= 2;
        return power2;
    }

    void loadImage(Scanner inFile){
        for (int row = 0; row < numRows; row++) {
            for (int col = 0; col < numCols; col++) {
                imgArray[row][col] = inFile.nextInt();
            }
```

```
        }
    }

    void zeroImgArray(){
        imgArray = new int[computePower2()][computePower2()];
    }

    QuadTreeNode buildQuadTree(int upperRow, int upperCol, int size,
BufferedWriter outFile) throws IOException {
        QuadTreeNode currentNode = new QuadTreeNode(upperRow, upperCol, size,
null, null, null,null);
        currentNode.printQuadTreeNode(outFile);
        if(size == 1)
            currentNode.color = imgArray[upperRow][upperCol];
        else{
            currentNode.NW = buildQuadTree(upperRow, upperCol, size/2,
outFile);
            currentNode.NE = buildQuadTree(upperRow, upperCol+size/2, size/2,
outFile);
            currentNode.SW = buildQuadTree(upperRow+size/2, upperCol, size/2,
outFile);
            currentNode.SE = buildQuadTree(upperRow+size/2, upperCol+size/2,
size/2, outFile);

            switch (sumKids(currentNode)) {
                case 0 -> {
                    currentNode.color = 0;
                    setLeaf(currentNode);
                }
                case 4 -> {
                    currentNode.color = 1;
```

```
                    setLeaf(currentNode);
                }
                default -> currentNode.color = 5;
            }
        }
        return currentNode;
    }

    int sumKids(QuadTreeNode currentNode){
        return currentNode.NW.color + currentNode.NE.color +
currentNode.SW.color + currentNode.SE.color;
    }

    void setLeaf(QuadTreeNode currentNode){
        currentNode.NW = null;
        currentNode.NE = null;
        currentNode.SW = null;
        currentNode.SE = null;
    }

    boolean isLeaf(QuadTreeNode currentNode){
        if(currentNode.NW != null) return false;
        if(currentNode.NE != null) return false;
        if(currentNode.SW != null) return false;
        return currentNode.SE == null;
    }

    void preOrder(QuadTreeNode currentNode, BufferedWriter outFile) throws
IOException {
        currentNode.printQuadTreeNode(outFile);
```

```
        if(!isLeaf(currentNode)){
            preOrder(currentNode.NW, outFile);
            preOrder(currentNode.NE, outFile);
            preOrder(currentNode.SW, outFile);
            preOrder(currentNode.SE, outFile);
        }
    }

    void postOrder(QuadTreeNode currentNode, BufferedWriter outFile) throws
IOException{
        if(!isLeaf(currentNode)){
            preOrder(currentNode.NW, outFile);
            preOrder(currentNode.NE, outFile);
            preOrder(currentNode.SW, outFile);
            preOrder(currentNode.SE, outFile);
        }
        currentNode.printQuadTreeNode(outFile);
    }

    void printImgArray(BufferedWriter outFile) throws IOException{
        for(var row: imgArray){
            for(var pixel: row)
                outFile.write(pixel + " ");
            outFile.write("\n");
        }

    }
}
```

```
public class Main {


    public static void main(String[] args) throws IOException {
        Scanner inFile = new Scanner(new File(args[0]));
        BufferedWriter outFile1 = new BufferedWriter(new FileWriter(args[1],
true));
        BufferedWriter outFile2 = new BufferedWriter(new FileWriter(args[2],
true));


        int numRows, numCols, minVal, maxVal;
        numRows = inFile.nextInt();
        numCols = inFile.nextInt();
        minVal = inFile.nextInt();
        maxVal = inFile.nextInt();


        QuadTree quadTree = new QuadTree(numRows,numCols,minVal, maxVal);
        quadTree.loadImage(inFile);


        outFile2.write("Image Array\n\n");
        quadTree.printImgArray(outFile2);


        outFile2.write("\n\n\n\nBuild Quad Tree\n\n");
        quadTree.root =
quadTree.buildQuadTree(0,0,quadTree.computePower2(),outFile2);




        outFile1.write("PreOrder\n\n");
        quadTree.preOrder(quadTree.root, outFile1);


        outFile1.write("\n\n\n\nPostOrder\n\n");
```

```
        quadTree.postOrder(quadTree.root, outFile1);

        outFile1.close();
        outFile2.close();
        inFile.close();
    }
}
```

Image 1:

```
36 40 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 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 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 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 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 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 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 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 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 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 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 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

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

Outfile 1 for Image 1:

PreOrder

```
5, 0, 0, 5500
5, 0, 0, 5055
5, 0, 0, 5050
5, 0, 0, 0011
```

0, 0, 0, nullnullnullnull

0, 0, 4, nullnullnullnull

1, 4, 0, nullnullnullnull

1, 4, 4, nullnullnullnull

0, 0, 8, nullnullnullnull

5, 8, 0, 0011

0, 8, 0, nullnullnullnull

0, 8, 4, nullnullnullnull

1, 12, 0, nullnullnullnull

1, 12, 4, nullnullnullnull

0, 8, 8, nullnullnullnull

0, 0, 16, nullnullnullnull

5, 16, 0, 0010

0, 16, 0, nullnullnullnull

0, 16, 8, nullnullnullnull

1, 24, 0, nullnullnullnull

0, 24, 8, nullnullnullnull

5, 16, 16, 1510

1, 16, 16, nullnullnullnull

5, 16, 24, 1100

1, 16, 24, nullnullnullnull

1, 16, 28, nullnullnullnull

0, 20, 24, nullnullnullnull

0, 20, 28, nullnullnullnull

1, 24, 16, nullnullnullnull

0, 24, 24, nullnullnullnull

5, 0, 32, 0050

0, 0, 32, nullnullnullnull

0, 0, 48, nullnullnullnull

5, 16, 32, 1010

1, 16, 32, nullnullnullnull

0, 16, 40, nullnullnullnull

1, 24, 32, nullnullnullnull

0, 24, 40, nullnullnullnull

0, 16, 48, nullnullnullnull

0, 32, 0, nullnullnullnull

0, 32, 32, nullnullnullnull

PostOrder

5, 0, 0, 5055

5, 0, 0, 5050

5, 0, 0, 0011

0, 0, 0, nullnullnullnull

0, 0, 4, nullnullnullnull

1, 4, 0, nullnullnullnull

1, 4, 4, nullnullnullnull

0, 0, 8, nullnullnullnull

5, 8, 0, 0011

0, 8, 0, nullnullnullnull

0, 8, 4, nullnullnullnull

1, 12, 0, nullnullnullnull

1, 12, 4, nullnullnullnull

0, 8, 8, nullnullnullnull

0, 0, 16, nullnullnullnull

5, 16, 0, 0010

0, 16, 0, nullnullnullnull

0, 16, 8, nullnullnullnull

1, 24, 0, nullnullnullnull

0, 24, 8, nullnullnullnull

5, 16, 16, 1510

1, 16, 16, nullnullnullnull

5, 16, 24, 1100

1, 16, 24, nullnullnullnull

1, 16, 28, nullnullnullnull

0, 20, 24, nullnullnullnull

0, 20, 28, nullnullnullnull

1, 24, 16, nullnullnullnull

0, 24, 24, nullnullnullnull

5, 0, 32, 0050

0, 0, 32, nullnullnullnull

0, 0, 48, nullnullnullnull

5, 16, 32, 1010

1, 16, 32, nullnullnullnull

0, 16, 40, nullnullnullnull

1, 24, 32, nullnullnullnull

0, 24, 40, nullnullnullnull

0, 16, 48, nullnullnullnull

0, 32, 0, nullnullnullnull

0, 32, 32, nullnullnullnull

5, 0, 0, 5500

PreOrder

5, 0, 0, 5500

5, 0, 0, 5055

5, 0, 0, 5050

5, 0, 0, 0011

0, 0, 0, nullnullnullnull

0, 0, 4, nullnullnullnull

1, 4, 0, nullnullnullnull

1, 4, 4, nullnullnullnull

0, 0, 8, nullnullnullnull

5, 8, 0, 0011

0, 8, 0, nullnullnullnull

0, 8, 4, nullnullnullnull

1, 12, 0, nullnullnullnull

1, 12, 4, nullnullnullnull

0, 8, 8, nullnullnullnull

0, 0, 16, nullnullnullnull

5, 16, 0, 0010

0, 16, 0, nullnullnullnull

0, 16, 8, nullnullnullnull

1, 24, 0, nullnullnullnull

0, 24, 8, nullnullnullnull

5, 16, 16, 1510

1, 16, 16, nullnullnullnull

5, 16, 24, 1100

1, 16, 24, nullnullnullnull

1, 16, 28, nullnullnullnull

0, 20, 24, nullnullnullnull

0, 20, 28, nullnullnullnull

1, 24, 16, nullnullnullnull

0, 24, 24, nullnullnullnull

5, 0, 32, 0050

0, 0, 32, nullnullnullnull

0, 0, 48, nullnullnullnull

5, 16, 32, 1010

1, 16, 32, nullnullnullnull

0, 16, 40, nullnullnullnull

1, 24, 32, nullnullnullnull

0, 24, 40, nullnullnullnull

0, 16, 48, nullnullnullnull

0, 32, 0, nullnullnullnull

0, 32, 32, nullnullnullnull

PostOrder

5, 0, 0, 5055

5, 0, 0, 5050

5, 0, 0, 0011

0, 0, 0, nullnullnullnull

0, 0, 4, nullnullnullnull

1, 4, 0, nullnullnullnull

1, 4, 4, nullnullnullnull

0, 0, 8, nullnullnullnull

5, 8, 0, 0011

0, 8, 0, nullnullnullnull

0, 8, 4, nullnullnullnull

1, 12, 0, nullnullnullnull

1, 12, 4, nullnullnullnull

0, 8, 8, nullnullnullnull

0, 0, 16, nullnullnullnull

5, 16, 0, 0010

0, 16, 0, nullnullnullnull

0, 16, 8, nullnullnullnull

1, 24, 0, nullnullnullnull

0, 24, 8, nullnullnullnull

5, 16, 16, 1510

1, 16, 16, nullnullnullnull

5, 16, 24, 1100

1, 16, 24, nullnullnullnull

1, 16, 28, nullnullnullnull

0, 20, 24, nullnullnullnull

0, 20, 28, nullnullnullnull

1, 24, 16, nullnullnullnull

0, 24, 24, nullnullnullnull

5, 0, 32, 0050

0, 0, 32, nullnullnullnull

0, 0, 48, nullnullnullnull

5, 16, 32, 1010

1, 16, 32, nullnullnullnull

0, 16, 40, nullnullnullnull

1, 24, 32, nullnullnullnull

0, 24, 40, nullnullnullnull

0, 16, 48, nullnullnullnull

0, 32, 0, nullnullnullnull

0, 32, 32, nullnullnullnull

5, 0, 0, 5500

Image 2:

40 64 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 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

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 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

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 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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

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 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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

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 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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

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 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 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 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 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 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 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 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 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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
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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 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 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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
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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 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 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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
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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 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 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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
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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 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 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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
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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 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 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Outfile 1 for Image 2:

PreOrder

5, 0, 0, 5555

5, 0, 0, 0555

0, 0, 0, nullnullnullnull

5, 0, 16, 0051

0, 0, 16, nullnullnullnull

0, 0, 24, nullnullnullnull

5, 8, 16, 0505

0, 8, 16, nullnullnullnull

5, 8, 20, 0101

0, 8, 20, nullnullnullnull

1, 8, 22, nullnullnullnull

0, 10, 20, nullnullnullnull

1, 10, 22, nullnullnullnull

0, 12, 16, nullnullnullnull

5, 12, 20, 0101

0, 12, 20, nullnullnullnull

1, 12, 22, nullnullnullnull

0, 14, 20, nullnullnullnull

1, 14, 22, nullnullnullnull

1, 8, 24, nullnullnullnull

5, 16, 0, 5100

5, 16, 0, 0101

0, 16, 0, nullnullnullnull

1, 16, 4, nullnullnullnull

0, 20, 0, nullnullnullnull

1, 20, 4, nullnullnullnull

1, 16, 8, nullnullnullnull

0, 24, 0, nullnullnullnull

0, 24, 8, nullnullnullnull

5, 16, 16, 5001

5, 16, 16, 1010

1, 16, 16, nullnullnullnull

0, 16, 20, nullnullnullnull

1, 20, 16, nullnullnullnull

0, 20, 20, nullnullnullnull

0, 16, 24, nullnullnullnull

0, 24, 16, nullnullnullnull

1, 24, 24, nullnullnullnull

5, 0, 32, 0550

0, 0, 32, nullnullnullnull

5, 0, 48, 0110

0, 0, 48, nullnullnullnull

1, 0, 56, nullnullnullnull

1, 8, 48, nullnullnullnull

0, 8, 56, nullnullnullnull

5, 16, 32, 0010

0, 16, 32, nullnullnullnull

0, 16, 40, nullnullnullnull

1, 24, 32, nullnullnullnull

0, 24, 40, nullnullnullnull

0, 16, 48, nullnullnullnull

5, 32, 0, 0500

0, 32, 0, nullnullnullnull

5, 32, 16, 0100

0, 32, 16, nullnullnullnull

1, 32, 24, nullnullnullnull

0, 40, 16, nullnullnullnull

0, 40, 24, nullnullnullnull

0, 48, 0, nullnullnullnull

0, 48, 16, nullnullnullnull

5, 32, 32, 5000

5, 32, 32, 1000

1, 32, 32, nullnullnullnull

0, 32, 40, nullnullnullnull

0, 40, 32, nullnullnullnull

0, 40, 40, nullnullnullnull

0, 32, 48, nullnullnullnull

0, 48, 32, nullnullnullnull

0, 48, 48, nullnullnullnull

PostOrder

5, 0, 0, 0555

0, 0, 0, nullnullnullnull

5, 0, 16, 0051

0, 0, 16, nullnullnullnull

0, 0, 24, nullnullnullnull

5, 8, 16, 0505

0, 8, 16, nullnullnullnull

5, 8, 20, 0101

0, 8, 20, nullnullnullnull

1, 8, 22, nullnullnullnull

0, 10, 20, nullnullnullnull

1, 10, 22, nullnullnullnull

0, 12, 16, nullnullnullnull

5, 12, 20, 0101

0, 12, 20, nullnullnullnull

1, 12, 22, nullnullnullnull

0, 14, 20, nullnullnullnull

1, 14, 22, nullnullnullnull

1, 8, 24, nullnullnullnull

5, 16, 0, 5100

5, 16, 0, 0101

0, 16, 0, nullnullnullnull

1, 16, 4, nullnullnullnull

0, 20, 0, nullnullnullnull

1, 20, 4, nullnullnullnull

1, 16, 8, nullnullnullnull

0, 24, 0, nullnullnullnull

0, 24, 8, nullnullnullnull

5, 16, 16, 5001

5, 16, 16, 1010

1, 16, 16, nullnullnullnull

0, 16, 20, nullnullnullnull

1, 20, 16, nullnullnullnull

0, 20, 20, nullnullnullnull

0, 16, 24, nullnullnullnull

0, 24, 16, nullnullnullnull

1, 24, 24, nullnullnullnull

5, 0, 32, 0550

0, 0, 32, nullnullnullnull

5, 0, 48, 0110

0, 0, 48, nullnullnullnull

1, 0, 56, nullnullnullnull

1, 8, 48, nullnullnullnull

0, 8, 56, nullnullnullnull

5, 16, 32, 0010

0, 16, 32, nullnullnullnull

0, 16, 40, nullnullnullnull

1, 24, 32, nullnullnullnull

0, 24, 40, nullnullnullnull

0, 16, 48, nullnullnullnull

5, 32, 0, 0500

0, 32, 0, nullnullnullnull

5, 32, 16, 0100

0, 32, 16, nullnullnullnull

1, 32, 24, nullnullnullnull

0, 40, 16, nullnullnullnull

0, 40, 24, nullnullnullnull

0, 48, 0, nullnullnullnull

0, 48, 16, nullnullnullnull

5, 32, 32, 5000

5, 32, 32, 1000

1, 32, 32, nullnullnullnull

0, 32, 40, nullnullnullnull

0, 40, 32, nullnullnullnull

0, 40, 40, nullnullnullnull

0, 32, 48, nullnullnullnull

0, 48, 32, nullnullnullnull

0, 48, 48, nullnullnullnull

5, 0, 0, 5555